

Review

Resource Management in SDN-Based Cloud and SDN-Based Fog Computing: Taxonomy Study

Amirah Alomari *, Shamala K. Subramaniam *, Normalia Samian, Rohaya Latip and Zuriati Zukarnain

Department of Communication Technology and Networking, Faculty of Computer Science, University Putra Malaysia, Selangor 43400, Malaysia; normalia@upm.edu.my (N.S.); rohayalt@upm.edu.my (R.L.); zuriati@upm.edu.my (Z.Z.)

* Correspondence: gs55300@student.upm.edu.my (A.A.); shamala_ks@upm.edu.my (S.K.S.)

Abstract: Software-defined networks (SDN) is an evolution in networking field where the data plane is separated from the control plane and all the controlling and management tasks are deployed in a centralized controller. Due to its features regarding ease management, it is emerged in other fields such as cloud and fog computing in order to manage asymmetric communication across nodes, thus improving the performance and reducing the power consumption. This study focused on research that were conducted in SDN-based clouds and SDN-based fogs. It overviewed the important contributions in SDN clouds in terms of improving network performances and energy optimization. Moreover, state-of-the-art studies in SDN fogs are presented. The features, methods, environment, dataset, simulation tool and main contributions are highlighted. Finally, the open issues related to both SDN clouds and SDN fogs are defined and discussed.

Keywords: software-defined networks; fog computing; cloud computing; energy efficiency; vehicular networks; edge computing



Citation: Alomari, A.; Subramaniam, S.K.; Samian, N.; Latip, R.; Zukarnain, Z. Resource Management in SDN-Based Cloud and SDN-Based Fog Computing: Taxonomy Study. *Symmetry* **2021**, *13*, 734. <https://doi.org/10.3390/sym13050734>

Academic Editor: Theodore E. Simos

Received: 5 March 2021

Accepted: 3 April 2021

Published: 21 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Traditional networks rely on network devices to make forwarding and routing decisions by implementing hardware tables that are embedded to the device itself, such as a bridge and router. As well, traffic rules such as filtering and prioritizing are implemented locally in the device. However, software-defined networks (SDN) brought advancement in networking by addressing simplicity, as it aims to reduce the design complexity involved in implementing both the hardware and software of network devices.

The basic concept of SDN is to remove the controllability from network devices and allow one central device, i.e., control unit to control and manage. The control unit is capable of observing the entire network and making decisions regarding forwarding and routing, whereas the task of forwarding is handled by network hardware devices in addition to filtering and traffic prioritization [1]. OpenFlow is a well-known SDN design that follows its basic architecture of decoupling control from the data plane where the controller and switches are communicating through OpenFlow protocols. Switches, on the other hand, contain flow tables and flow entries such as matching fields, counters and sets of actions, while the controller is capable of performing sets of actions on flow entries, such as update, delete and add [2].

Even though cloud computing is a powerful technology, there are some issues and challenges that are still open, such as network mobility, scalability and security issues; thereafter, the possibility to extend software-defined networks in cloud computing has been investigated [3]. Although fog computing brought solutions to handle IoT applications that requires less latency and a higher bandwidth to process data from cloud server to end devices. However, end devices have limited resources that conflict with high demands, besides heterogeneity, which is another challenge introduced in fog computing [4]. Therefore, the current studies are developed for the sake of bringing advantages of SDN to both cloud

and fog computing in order to overcome the above-mentioned issues by offering flexible and intelligent resources management solutions.

Furthermore, resource management is a challenging issue, as the difficulty lies with its relatedness to several problems, such as resource heterogeneity, asymmetric communication, inconsistent workload and resources dependency [5,6]. Therefore, this study reviews related research and its contribution to SDN-based cloud in terms of network performance, and energy efficiency. Additionally, it presents state-of-the-art contributions in the field of emerging software defined networks and fog computing. This review contributes to two of the most promising fields in networking; its importance lies in providing a comprehensive review of important and state-of-the-art studies related to SDN cloud and SDN fog. Additionally, it helps researchers to observe related contributions by assessing relative important key information of a specific study. Thus, it can be used as reference for researchers based on their area of concern; since this review classifies novel findings and summarizes it based on several criteria that help researchers to identify, select and contribute to SDN-based cloud/fog accordingly. Moreover, each reviewed research is mapped with corresponding evaluated metrics for ease of configuration of correlated studies based on common interest.

The remainder of this review is organized as follows: Section 2 overviews the methodology of our literature review. Section 3 presents SDN-based cloud resources management solutions that contribute in network performance enhancements. While SDN-based cloud research that contribute in energy efficiency optimization are overviewed in Section 4. Furthermore, Section 5 shows state-of-the-art contributions in SDN-based fog. Section 6 presents open issues of both SDN-based clouds and SDN-based fogs. Finally, Section 7 presents the concluding remarks.

2. Methodology of the Literature Review

This section presents the methodology used for conducting our review. It follows a systematic review where the reviewed studies were either journal articles or conference papers published in IEEE Xplore, Springer, and ScienceDirect datasets. The keywords used for searching include: “SDN + cloud + resources + management”, “SDN + cloud + resources + allocation”, “SDN + cloud + Virtual Machine (VM) + placement” and “SDN + cloud + VM + migration”. As well, “SDN + fog + resources + management”, “SDN + fog + resources + allocation”, “SDN + fog + VM + placement” and “SDN + fog + VM + migration”. After that, results are filtered based on time period from 2019 to 2021. However, please note that we also reviewed the mostly mentioned baseline works related to the SDN cloud to be used as future references. After that, selected researches are analyzed thoroughly and summarized based on a major feature of the research, main tasks of the proposed algorithm, targeted environment of research, main contribution, evaluation tool, dataset and metrics used for evaluation.

3. SDN-Based Cloud Resource Management in Network Performance Improvement

Regarding network performance, there are many algorithms that target different aspects of network performance metrics, such as response time, bandwidth, throughput and other quality of service (QoS) measures in different environments. This section is subdivided based on the focus of the design of resource management algorithms to be either based on QoS constraints, priority-aware, and QoS-aware algorithms. The last section concentrates on VM migration algorithms.

3.1. Resources Management Based on QoS Constraints

The MAPLE system [7] is developed in order to manage resource allocation for both network and computing resources. It uses effective bandwidth estimation based on server's traffic traces analysis. Effective bandwidth estimation is used to specify needed bandwidth to process requests without a violation of QoS constraints. The system is composed of a centralized controller, which processes and manages requests, and effective bandwidth agents on servers. The controller performs preprocessing of VM requests by

defining QoS targets and resembling incoming requests to be placed as one VM, calculates residual bandwidth estimations and then places VM. After that, network-aware VM placement algorithm is executed, which allocates VM in first fit decreasing approach in order to address the problem of consolidation while achieving QoS constraints. The system is evaluated in emulated virtual datacenter and compared to two network resources algorithms which are Oktopus [8] based on expected mean and expected peak throughputs. Maple managed on achieving lower QoS constraint (delay) than the expected value, as well as allocating jointly resources efficiently. Since the MAPLE system succeeded on maintaining QoS constraints at lower rates than expected, an extension, MAPLEx [9], is developed to further incorporate anti-colocation problem. The algorithm takes an extra input that is number of VM slots that are preserved for a particular request for VM placement. The algorithm is evaluated in an extended testbed compared to MAPLE evaluation. As well, the dataset that used for evaluation consist of three types that are data intensive applications, data serving applications, and data backup applications. The results show that MAPLEx is able to attain satisfied QoS levels while locating network and cloud resources. In addition to MAPLEx, Maple is developed for SDN environments to be MAPLE-Scheduler [10], which is a built-in flow scheduling unit in SDN controller in order to maintain QoS constraints when it is triggered. It is developed for Maple systems in software defined networks. The main feature of Maple-scheduler is that it is capable of observing network status and, in the case of a QoS violation occurs, it reschedules flows dynamically using effective bandwidth estimation (EB) of the top of rack switches, and the effective bandwidth coefficient (EBC) estimation of core switches. Moreover, it uses monitoring agents that are distributed over network (servers and edge switches) in order to acknowledge MAPLE-scheduler regarding the topology of a network, as well as the flow used resources. Basically, flows rescheduling take place when there is violation in QoS for large flows, and the decision will be made based on existing residual bandwidth and maintain load fairness by selecting links with minimum residual bandwidth. Evaluation is conducted against the Equal-Cost Multi-Pathing protocol ECMP, which is a routing protocol that selects the best paths to forward a packet through [11]. The comparison is based on different QoS constraints scenarios. The results show that Maple-scheduler achieves lower QoS violation rates compared to ECMP, as well as maximizes the throughput in all scenarios.

On the other hand, establishing a fair charging model is the motivation behind developing Oktopus [8]. It is proposed to solve unpredictability in network performance in cloud environment; by pursuing a solution that guarantees both parties rights: providers and tenants from perspectives of cost, profits, and QoS constraints. Therefore, virtual network abstractions are established and offered to tenants instead of computing resources by selecting number of VMs, types of VMs, and virtual network connecting them. Hence, virtual network abstraction decouples tenants from infrastructure. Oktopus is an implemented system which is evaluated by simulations and testbed. The results show that datacenters can reduce the overall cost by charging for internal traffics in virtual networks which leads to reduced tenants' costs and maintained providers' profit while concurrently preserve performance. Furthermore, to solve the problem of big data processing, resource allocation algorithm is developed by combining SDN and virtual machines, specifically complex event processing machines CEP [12]. Individual CEP contains a load balancer that works dynamically based on data stream arrival. While a performance optimizer works as a load distributor between all CEPs. Resource allocation is performed by taking into consideration three factors, which are: size of data, time and virtual machine. After that, processing big data is performed by considering individual packets. Processing is composed of three parameters, which are: data packet deadline, amount of work, and required time to reach virtual processor. Finally, job scheduling is performed in FIFO manner. The evaluation is conducted in CloudSim simulation tool with SDN setting. Various types of data are used for testing and time is the evaluation metric. Results show the effectiveness of using CEP and SDN in reducing required time to process data stream.

3.2. Priority-Aware Resources Management

This subsection presents algorithms that focused on priority and methods to assign privilege to high priority requests, such as in [13], proposed a prioritization model in cloud environment that uses the crystalline mapping CM algorithm. Main idea is that multiple tenants initiate service requests with a declared priority information, then it will be sent to a central master server. The master server job will prioritize arrived request queues by considering several attributes, which are fairness and maximum revenue. Thus, each tenant will prioritize requests locally first as the initial phase by following the multivariate likelihood ratio algorithm and then sorts the requests based on their scores. Then, the master server will prioritize requests queues from multiple tenants by using the crystalline mapping algorithm, which preserves fairness, isolation and maintain requests orders. The algorithm will maintain requests priority queues by calculating global priority level. After that, from priority queues, the request of maximum estimated revenue is selected for processing. The framework is evaluated in simulation environment, and the evaluation results show that average waiting time, which was declared as fairness property, was almost the same for all processed requests. While requests of maximum revenue were processed first by CM model compared to baseline work.

Whereas, in [14], the authors developed a client's priority-based resources provisioning model, high or low, in which a cloud provider offers pool of resources for clients. However, some of these resources are provisioned for only high priority clients. Thus, cloud providers increase profit by maintain SLA for such clients whenever there is demand on service at any given time. On the other hand, low priority clients will get access to shared resources, which also will meet their SLA contract. The authors calculated rejection probability, which indicates the probability of rejected clients' requests in such a way that high priority clients supposed to get the least rejected requests compared to low priority clients. The proposed work was evaluated in simulation environment. They conclude that, the rejection probability shows that number of high priority clients rejected requests is ten times less than rejected requests of low priority clients.

As well, in [15], the resource allocation algorithm was proposed in the software-defined cloud environment SDCC, which considers an application priority as main consideration when placing VMs and provision network bandwidth. An application is previously determined to be either critical or normal in priority criterion. Submitted application requests contain information about VM and flow, which include number of processing cores, core's capacity, amount of memory, storage size, required bandwidth, source and destination VM, and application priority. However, when application requests are placed, the system will allocate host and network resources based on provided information. Firstly, the system will determine the network topology and hosts and VM grouping decisions is to be made based on hosts connectivity and application, respectively. Then mapping of VM to hosts is performed based on the highest amount of joint resources taking into consideration host connectivity to be either on single group or within closest proximity in order to minimize network traffic. If an application is characterized as critical, then network bandwidth is provisioned by SDN controller via priority queues on link switches. The proposed algorithm is compared with exclusive resource allocation, random allocation, and combination of fit first decreasing and dynamic flow algorithms [7] in the CloudSimSDN simulation tool against the response time and power consumption in two different scenarios (synthetic workload and Wikipedia workload). The proposed algorithm manages to reduce response time in both scenarios. While power consumption is reduced compared to most baseline algorithm for critical workloads. However, there is not significant reduction in power consumption in the case of Wikipedia workloads which indicates that the proposed algorithm is not power demanding. The proposed algorithm fits properly with the critical applications in synthetic workload for response time reduction since bandwidth provisioning algorithm is only triggered when network is intensive with workloads.

3.3. QoS-Aware Resources Management

While some algorithms consider priority as main challenge, others concentrate more on establishing QoS aware algorithms, such as the quality of service aware virtualization-enabled routing (QVR) algorithm [16]. It is proposed for SDN environment where isolated slices of physical infrastructure are created, and each slice is dedicated for several tenants. It aims to dynamically support flow allocation and provision end-to-end QoS constraints fulfillment. Performance evaluation shows that QVR capable of reducing number of shared links while accomplishing lowest link congestion rates compared to baselines. In addition to that, EQVMP [17] is the VM placement algorithm, which aims to preserve low power consumptions and QoS-aware method. It combines three techniques to achieve desirable results. The input consists of VM resource demands, topology matrix and VM traffic. After that, EQVMP will reduce number of hops by maintain same number of VMs in multiple groups to reduce cost of traffic load. Then, the VMs are to be sorted in decreasing order based on total resources needed, then mapping VMs to suitable servers take place by selecting a VM, which is best fit to a specific server. In case there is not an ON-server able to accommodate the VM demands, then a new server will be woken up. Finally, load balancing feature in SDN controller is utilized to prevent link congestion by selecting substituted links in case of overutilization. EQVMP achieves nearly optimal results in power savings, however, average delay is high. Nevertheless, it succeeds on achieving outstanding throughput compared to other baselines and this is the reason this algorithm is located under network performance section.

3.4. Contributions on VM Migration Task in Resources Management

Although most of above-mentioned algorithms focused on VM placement, others consider VM migration as part of resource management, which targets QoS such as in [18], authors developed a live VM migration, S-CORE in SDN environment, which is derived from S-CORE in clouds [19]. S-CORE takes into consideration colocation constraints in order to minimize overall communication cost by assigning links with the ratio of over-subscription, as well the cost of bandwidth as the weights. Then, using temporal bandwidth utilization multiplied with aggregate weightings, overall cost is obtained. The system architecture consists of server virtualization to manage VMs through an open-source API; Libvirt. While the software switch is Open vSwitch at the hypervisor level and SDN controller with OpenFlow specification. Moreover, the controller modules include topology discovery, host discovery, links weights, flow statistics and REST API with certain endpoints for controllers to migrate VM, depending on the network state. The proposed work is evaluated using simulation: NS3 and testbed, where, link utilization, overall communication cost, VM-VM communication cost, throughput, scalability, and Number of migrations are the evaluation metrics. The results show that there is a significant decrease in congestion, and the overall throughput is enhanced. Furthermore, this algorithm is proposed in clouds datacenters through tokens, which hold information such as the VM ID, communication level value to determine communication cost. As well, each VM has the token will have an advantage of determining the next VM to pass the token based on specific policy such as round robin.

While Remedy [20] is proposed to manage VM migration by managing steady states. It is composed of three components that are bandwidth monitor, which will send VMs list to VM and target selector that determines a host to migrate VMs and VM memory monitor. The system is evaluated in testbed using real datacenter traces that shows the effectiveness of the Remedy system in reducing the unsatisfied bandwidth.

The following table (Table 1) summarizes above mentioned network performance contributions in terms of major feature of algorithms, main tasks, the environments where the algorithm targets, which evaluation tool is used for testing and the dataset, evaluated metrics and, finally, the major contribution of the algorithm based on its published results. In addition to that, Table 2, declares evaluation metrics mapped with corresponding research.

Table 1. Summarization table of contributions in network performance.

Ref.	Major Feature	Main Tasks	Env.	Dataset	Eval. Tool	Measured Metrics	Contribution
[7]	<ul style="list-style-type: none"> Empirical estimation of effective bandwidth using MAPLE controller and EB agents. 	<ul style="list-style-type: none"> VM ensemble placement based on effective bandwidth—VMs are placed on same server (MAPLE) 	DCs	<ul style="list-style-type: none"> 300MB dataset as bulk data. Tenants requests 	Emulated virtual DCN	<ul style="list-style-type: none"> QoS overall and average violation rate: proportion of delayed packets. 	<ul style="list-style-type: none"> Placing new VMs while maintaining bandwidth limits Preserve QoS targets under violation rate
[8]	<ul style="list-style-type: none"> Virtual Networks abstractions 	<ul style="list-style-type: none"> Predictable performance using virtual network abstraction Oktopus system 	<ul style="list-style-type: none"> Cloud datacenters Production datacenters 	<ul style="list-style-type: none"> Data-intensive application Local communication pattern application 	Simulation and testbed	<ul style="list-style-type: none"> Job completion time Utilization Rejected requests Cost and profit 	<ul style="list-style-type: none"> Fair charging model
[9]	<ul style="list-style-type: none"> Empirical estimation of effective bandwidth 	<ul style="list-style-type: none"> Enable ensemble VM placement with anti-colocation constraint (MAPLEx) 	DCs	<ul style="list-style-type: none"> Data backup application: based on SCP Data analysis application: word count, Hadoop. Data serving application: YCSB ad Cassandra as data client and server 	Testbed	<ul style="list-style-type: none"> QoS violation: overall and average. Reject rate of ensemble placement requests. Link utilization. 	<ul style="list-style-type: none"> Similar QoS violation rate with MAPLE with more rejection rate thus lower mean link utilization.
[10]	<ul style="list-style-type: none"> Centralized Flow scheduler using effective bandwidth with distributed monitoring agents Joint Allocation 	<ul style="list-style-type: none"> VM placement: observe network changes Reschedule flows dynamically Maintain QoS of VM. Preserve link utilization (MAPLE Scheduler) 	SDN	<ul style="list-style-type: none"> Data applications as bulk data transfer: word count Hadoop, data serving YCSB and SCP. Custom data backup applications. 	Testbed	<ul style="list-style-type: none"> QoS violation rate: overall and average Throughput Link utilization 	<ul style="list-style-type: none"> Significant decrease of overall QoS violation. Better median throughput

Table 1. Cont.

Ref.	Major Feature	Main Tasks	Env.	Dataset	Eval. Tool	Measured Metrics	Contribution
[12]	<ul style="list-style-type: none"> Use of complex event processing machines 	<ul style="list-style-type: none"> Resource allocation, processing of big data stream, job scheduling, and load balancing 	SDN-Cloud	<ul style="list-style-type: none"> Different types of data. 	Cloudsim with SDN settings	<ul style="list-style-type: none"> Time 	<ul style="list-style-type: none"> Effectiveness of both CEP and SDN in reducing time required to process data
[13]	<ul style="list-style-type: none"> Considers Local & global priority Prioritization model based on crystalline mapping CM algorithm 	<ul style="list-style-type: none"> Statistical information model Local priority model Global crystalline mapping model Maximum Revenue Evaluation. 	<ul style="list-style-type: none"> Clouds 	<ul style="list-style-type: none"> Real world data collection 	Simulation	<ul style="list-style-type: none"> Response Time Revenue 	<ul style="list-style-type: none"> Greater revenue with maintained response time
[14]	<ul style="list-style-type: none"> Rejection Probability of two priority requests: high & low. 	<ul style="list-style-type: none"> Resource provisioning for different priority requests 	<ul style="list-style-type: none"> Clouds 	<ul style="list-style-type: none"> Based on number of servers = 40,000. 	Simulation	<ul style="list-style-type: none"> Rejection probabilities for different priority requests 	<ul style="list-style-type: none"> Analytical determination of rejection probability for different priority classes for resources allocation problem.
[15]	<ul style="list-style-type: none"> Local prioritizing of requests (normal/critical) Heterogeneous Joint allocation 	<ul style="list-style-type: none"> VM allocation Bandwidth provision 	<ul style="list-style-type: none"> SDN-Cloud Computing 	<ul style="list-style-type: none"> Synthetic workload Wikipedia workload 	CloudSimSDN	<ul style="list-style-type: none"> Power Consumption Response time 	<ul style="list-style-type: none"> Reduced response time
[16]	<ul style="list-style-type: none"> Adaptive feedback management tool: QVR 	<ul style="list-style-type: none"> Jointly virtualization and routing technique with Dynamic flow allocation based on QoS 	<ul style="list-style-type: none"> SDN 	<ul style="list-style-type: none"> Real time and non-real time application 	Simulation	<ul style="list-style-type: none"> Delay Latency Number of shared links 	<ul style="list-style-type: none"> Better tenants' isolation with enhanced congestion latency and less delay.
[17]	<ul style="list-style-type: none"> Three-tier algorithm: Hop reduction Power saving Load balancing 	<ul style="list-style-type: none"> Virtual Machine placement considers QoS and power saving: EQVMP 	<ul style="list-style-type: none"> DCN 	<ul style="list-style-type: none"> Not available 	Simulation: NS2	<ul style="list-style-type: none"> Power consumption Average delay Throughput 	<ul style="list-style-type: none"> Significantly increase of system throughput while acceptably decrease power consumption and average delay.

Table 1. Cont.

Ref.	Major Feature	Main Tasks	Env.	Dataset	Eval. Tool	Measured Metrics	Contribution
[18,19]	<ul style="list-style-type: none"> Assigning each link with a weight metric Using colocation and network locality 	<ul style="list-style-type: none"> Live VM migration decoupled from SDN controller: S-CORE 	<ul style="list-style-type: none"> SDN Cloud DC 	<ul style="list-style-type: none"> Traffic is generated by Nping tool in SDN. DC traffic generator in clouds. 	<ul style="list-style-type: none"> Testbed for SDN Simulation: NS3 for Cloud DCs.	<ul style="list-style-type: none"> Link utilization Overall communication cost and VM-VM communication cost Throughput Scalability Number of migrations 	Significant decrease in congestion and enhance overall throughput.
[20]	<ul style="list-style-type: none"> Steady state management 	<ul style="list-style-type: none"> VM placement: bandwidth monitor, target selector, and VM memory monitor: Remedy 	<ul style="list-style-type: none"> DCN 	<ul style="list-style-type: none"> Real datacenter traces 	Simulation	<ul style="list-style-type: none"> Number of VM migration Percentage reduction in unsatisfied bandwidth Link utilization 	Reduce unsatisfied bandwidth

4. SDN-Based Cloud Computing Resource Management in Energy Efficiency

Many techniques were employed in order to minimize the power consumption. For example, in [21], the authors used dynamic overbooking of resources in software-defined data centers; dynamic overbooking is claimed to reduce power consumption for both hosts and network resources. Hosts and network resources are overbooked using a dynamic ratio that is determined based on current workloads, which results in the obligation of SLAs. SLA violation was measured based on response time, such that each request is measured in advance to obtain the response time under no overbooking strategy. After that, with proposed overbooking strategy, the number of requests that exceed the response time, which was previously measured, will be considered as a violation of the SLA. The performance is evaluated in the CloudSimSDN simulation tool using Wikipedia workloads as datasets. Extensive experiments were conducted that show that dynamic overbooking has an advantage over static overbooking in terms of decreasing the SLA violation rate and reduced power consumption.

On the other hand, FCTcon [22] deploys a flow completion time FCT for dynamic flow management especially for delay sensitive applications. It aims to decrease the power consumption in data center networks while guarantee the required FCT. It considers the burst nature of the flow. Basically, FCTcon will receive requests with specified bandwidth requirements. Then, it will send it, with control knob = 1 to have their proposed bandwidth requirement, to traffic the consolidation unit that will determine what flow will be used, and the rest of flows will be set into sleep mode. After that, the FCT monitor unit will observe the FCT of requests and then acknowledge FCTcon controller in order to modify the control knob that will either be increased or decreased value, which, in turn, will indicate number of flows to be either wake up or sleep by traffic consolidation unit in order to accommodate the FCT. The proposed work is evaluated in testbed environment using Wikipedia and Yahoo traffic files. The results show that FCTcon achieves the required request file completion time. Even though it succeeds on coping with the optimal power consumption levels, some of baselines manage to reduce power consumption below the optimal levels. Additionally, PowerFTC [23] works by considering FCT, especially for delay-sensitive flows in order to reduce the power consumption. In order to impose FCT during the consolidation process, it obtains flow paths and switch settings in advance. The evaluation, which is conducted in testbed and simulation, show that, loosely, FCT requirements lead to a higher power saving while the FCT miss ratio is the minimum in all cases.

While DISCO [24] is developed to optimize power consumption in the data center networks by utilizing traffic consolidation, it offers cloud providers with two algorithms, DISCO-F and DISCO-S, to be chosen from based on the most desirable objectives which are power efficiency, scalability and enhancing the delay performance of the data center network. Both Disco-F and Disco -S perform consolidations; however, the difference lies in where the consolidation takes place. DISCO-F performs individual flow consolidation, while DISCO-S performs multiple flow consolidation on individual switches. Using Wikipedia and Yahoo traces, the evaluation results show that, in terms of delay, DISCO algorithms manage to obtain lower delay than most of baseline algorithms. However, regarding power saving, both algorithms consume more power than most baseline algorithms.

Additionally, Elastic Tree [25] is a power optimizer that switches unused resources ON or OFF based on its current usability by monitoring traffic. Additionally, it targets fault tolerant and performance when making the decision of which subnet is supposed to be ON. It works by receiving traffic status by the optimizer, which is implemented with various methods depending on providers' goals: power saving, performance and fault tolerant. After that, a subnet of topology and flow path are sent to control the software. The system is evaluated in the testbed and proved its capability of supporting performance and robustness while reducing the power consumption.

In addition to that, by analyzing traffic bandwidth usage in datacenter networks, CARPO [26] is proposed to reduce the power consumption through utilizing the minimum

subnet of the network. By tracing Wikipedia and Yahoo traces, various traffic peak time and correlation are observed. This leads to conclude that correlation is either to be loosely or negative and consolidation of traffic value is suggested to be designed, depending on the on and off peak values. Therefore, CARPO works by analyzing correlation from previous period, then consolidates traffic based on link capacity and, finally, adapts the link rates. It is evaluated based on power saving, power consumption, packet drop ratio and delay using testbed and simulation. Results show that delay and packet drop ratio are considered higher than baselines; however, CARPO succeeds in power saving.

Moreover, DCTCP is proposed TCP-like protocol that is specified for datacenter networks [27]. It works by estimating cognition degree from fraction of marked packets extracted from Explicit Congestion Notification (ECN). Thus, DCTCP enrich ECN with control schema in order to reduce buffer space while maintain network performance. While a study showed that a scalable network topology; flattened butterfly with commercial switch chips provides full utilization with reduced energy proportional datacenter [28]. Whereas, in [29], VM placement is addressed by combining two approaches that are genetic and the tabue search algorithms to reduce power consumption in cloud environment. Tabue search algorithm is used to enhance genetic algorithm searching by performing mutation operations. GATA algorithm is evaluated using PyCharm with Google Cluster dataset. Results show that GATA manages to reduce power consumption while lowering load balance. Additionally, it gives highest number of optimal solutions for different size problems. However, execution time is affected as it falls in the third-best position compared to four baselines.

As well, HGAPSO is a multi-objective resource allocation based on Euclidean distance, which combines two algorithms: Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) for VM placement. It comprises three main phases: VM allocation, task scheduling, and VM migration. It focused on reducing the power consumption while increasing the resource utilization. The results of evaluation show that HGAPSO consumes more energy and less resources (CPU, RAM and storage). However, it works much better when combined with VM migration in term of both energy consumption and resource utilization [30]. Moreover, multi-objective virtual cluster allocation framework is proposed in [31] based on four tradeoff models, which are VM availability, saving energy, resource utilization and load balance. Then IBBBO algorithm is developed based on natural evolutionary algorithm in which weights are assigned for previous models as a tradeoff relationship. It is tested using CloudSim where power consumption, average resources utilization, risk value and load balance value are measured. Results show that IBBBO manages in obtaining lower risk value, less power consumption, better average resources utilization with the least load balance value compared to most of the baselines.

In addition to that, several algorithms are proposed to find the acceptable balance between energy consumption and SLA violation, such as in [32], a framework for resource management that considers VM placement in such a way that SLA is maintained, and the energy consumption is reduced. It is composed of three modules responsible of cluster management including VM placement, overload detection of each PM CPU and resource prediction for each VM. The number of overloads and number of SLA violation, and power consumption are evaluated. Results show that a smaller number of overloads occurred with an outperforming on minimizing SLA violation. While for power consumption is almost similar with baseline. While a simple linear regression prediction model RobustSLR [33] is proposed to determine the utilization of server's resources (CPU) and verifies whether it is underloaded or overloaded, as well predicts future utilization of resources. Moreover, eight migration policies are proposed and evaluated. It aims to reduce power consumption as well SLA violation. evaluation is conducted in CloudSim simulation tool with two types of workloads: random and real world (PlanetLab data). Power consumption, SLA violation, overall and average SLA violation are measured. Most proposed algorithms consume less power while violating less SLA compared to baselines. However, one algorithm is an

exception, although, it consumes similar amount of power energy compared with baselines but outperforms in reducing SLA violation.

Similarly, predictive anticorrelated VM placement algorithm PACPA [34], which considers predictive resources prior to VM migration such that future demand is investigated in advance. It finds the suitable balance in terms of CPU utilization in order to perform VM placement. Therefore, two VMs are placed if their combined CPU requirement match the maximum demand of a specific host in order to utilize resources. Moreover, the algorithm uses neural networks as predictive technique in order to dynamically investigate future VM placements. PACPA is evaluated in CloudSim based on energy consumption and SLA violation. It accomplishes best results regarding reducing power consumption with smaller SLA violation compared to baselines. In [35], VM consolidation algorithms are proposed in order to reduce power consumption by taking into consideration not only CPU as a major power consumption element but, also, RAM. The proposed work consists of two managers: global and centralized manager and server level local manager. Then, two algorithms are developed for VM selection and placement to avoid overloaded servers to reduce power consumption as well reduce SLA violation. Moreover, it utilized two thresholds to detect overloaded and underloaded hosts. An evaluation is conducted using CloudSim simulation tool considering power consumption, SLA violation, and performance degradation after migration with static and dynamic threshold as evaluation metrics. The results show that the dynamic threshold brought better results to proposed algorithms especially with power consumption and performance degradation after migration. However, the SLA violation rate is higher than baselines in both thresholds.

On the other hand, some research focused only on power consumption. Several experiments of different policies: FCFS; SJF; different power management techniques (non-power aware NPA, power aware PA, dynamic voltage and frequency scaling DVFS), as well as different power models (linear, square, cubic and square root) are conducted in [36]. The results show that DVFS techniques consume the least power in both allocation policies FCFS and SJF. Moreover, more evaluation is conducted under different power models and results show that cubic model saves more power. Therefore, two algorithms are proposed based on experimental findings that are DFCFS and DVFS. Comparison between the two proposed algorithms is performed which show that DVFS consumes less power than DFCFS. Whilst PDM [37] is a centralized manager developed for power distribution in cloud datacenters by implementing a minimization cost function, linear optimization model, for uninterruptable power supplies UPS. The manager collects power consumption needed for resources (CPU) from each server through the server power module. Additionally, the local power controller is responsible on imposing server power threshold, both modules are server resides agents. Evaluation is conducted in testbed. The results show that a power assignment applied properly in such a way that it prevents UPSs overloading.

As well, a three-phase framework is proposed in order to manage virtual resources. Profiling and profile updating, which aim to process data and obtain information from and about the tasks, VM and PM [38]. Task classification to three classes that are long, normal and tiny. Then, the application assignment of these classified tasks to appropriate VMs is conducted. Finally, successive virtual machine placement is done by utilizing the pm that are most energy efficient. Three different scales that are large, medium and small datacenter are evaluated with the proposed work based on number of VMs and PM quantity gaps. As well a power saving is measured which result in increased by at most 12% compared to baseline. Moreover, the VM placement algorithm based on genetic algorithm was developed in [39]. The controller in the proposed system takes into consideration cost and computing requirements in order to determine the suitable VM selection type, which is performed through selection manager module. Additionally, the controller will determine the suitable placement via host selection manager module considering the objective of reducing cost and power consumption. However, in order to achieve that, an adaptive genetic algorithm is proposed which perform VM consolidation

in lowest number of servers as possible. The proposed work is evaluated in CloudSimSDN with synthetic and real workloads. The power consumption of network devices, power consumption of servers, average response time of both workloads, traffic load, and hourly cost are the evaluation metrics. Results show that proposed algorithm outperforms in reducing power consumption, response time and hourly cost.

Table 3 summarizes above-mentioned power optimization contributions in terms of major feature of algorithms, main tasks, the environments where the algorithm targets, which evaluation tool is used for testing and the dataset, evaluated metrics and, finally, the major contribution of the algorithm based on its published results. Moreover, Table 4 shows measured metrics with associated algorithms.

Table 3. Summarization of Power Optimization Contributions.

Ref.	Major Feature	Main Tasks	Envi.	Dataset	Evaluation Tool	Measured Metrics	Contribution
[21]	Dynamic overbooking & correlation analysis	Overbooking controller: VM placement & flow consolidation using overbooking ratio and correlation analysis. Resource Utilization Monitor: collects history data of PM utilization metrics, VM & Virtual links.	SDN- clouds	Wikipedia workload	Simulation: CloudSimSDN	<ul style="list-style-type: none"> Response time Power consumption 	Reduced power consumption while reducing SLA violation rate.
[22]	<ul style="list-style-type: none"> Traffic consolidation which Considers flow completion time requirement. considers delay-sensitive flows 	Traffic consolidation method that consider file completion time: FCTcon	DCN	Real DCN traces from Wikipedia and Yahoo.	Simulation & testbed	<ul style="list-style-type: none"> FCT Power saving 	<ul style="list-style-type: none"> Reduced power consumption maintain file completion time FTC constraint.
[23]	<ul style="list-style-type: none"> File completion time for delay sensitive flows 	Power optimization: PowerFCT	DCN	Wikipedia DCN traces.	Testbed and simulation	<ul style="list-style-type: none"> Power saving FCT miss ratio 	Loosely FCT requirements lead to higher power saving while FCT miss ratio is the minimum in all cases
[24]	<ul style="list-style-type: none"> Distributed traffic consolidation for individual flows for scalability purposes. Distributed traffic flows consolidation on individual switches for power saving purposes. 	<ul style="list-style-type: none"> Traffic flow consolidation by implementing an optimizer either on servers or switches depends on DCN purposes; power saving or scalability: DISCO-F & DISCO-S algorithms. 	DCN	Real DCN traces from Wikipedia, Yahoo, and DCP.	Simulation & testbed.	<ul style="list-style-type: none"> Power saving Delay 	<ul style="list-style-type: none"> Power saving with reduced computation overhead.
[25]	Using traffic stats and control over flow path.	Power optimization: Elastic Tree	DCN	Traffic generator tool.	Testbed	<ul style="list-style-type: none"> Power saving Robustness Latency 	Ability to reduce power while support performance and robustness.

Table 3. Cont.

Ref.	Major Feature	Main Tasks	Envi.	Dataset	Evaluation Tool	Measured Metrics	Contribution
[26]	Dynamic consolidation of traffic flow based on correlation analysis.	Power optimization: correlation analysis, traffic consolidation, and link rate adaptation: CARPO	DCN	Wikipedia and Yahoo traces.	Testbed	<ul style="list-style-type: none"> Power consumption Power saving Delay Packet drop ratio 	Outperform in power saving.
[27]	Early deduction of congestion	Reducing buffer space while maintain performance by empowering Explicit Congestion Notification (ECN): DCTCP	DCN	Three types of traffic: <ul style="list-style-type: none"> Query Short-message Background 	Testbed	<ul style="list-style-type: none"> Throughput Queue length convergence query delay timeout queries flow and query completion time 	
[28]	Butterfly topology to reach scalable network, full utilization with reduced power		DCN	<ul style="list-style-type: none"> Synthetic workloads and production datacenter Traces: uniform and advert workloads.	Simulation	<ul style="list-style-type: none"> Fraction of time spent at each link Latency 	Links operate mostly in low power consumption mode
[29]	Genetic algorithm with the tabue search algorithm	VM placement using Genetic Algorithm where Tabue search algorithm is used to enhance search by performing mutation operations.	Cloud Datacenter	<ul style="list-style-type: none"> Google Cluster Dataset 	Pycharm 3.3	<ul style="list-style-type: none"> Load balance Power Consumption Number of optimal solutions Execution time 	Reduced power consumption while lowers load balance. Also gives highest number of optimal solutions.
[30]	Multi-objective resources allocateon based on Euclidean distance uses Genetic Algorithm (GA) and Particle Swarm Optimization (PSO)	<ul style="list-style-type: none"> VM allocation Task scheduling VM migration	CloudDC	<ul style="list-style-type: none"> Heterogenous and homogenous DCs. 	Java	<ul style="list-style-type: none"> Energy consumption Resource utilization 	Better results when combined with VM migration

Table 3. Cont.

Ref.	Major Feature	Main Tasks	Envi.	Dataset	Evaluation Tool	Measured Metrics	Contribution
[31]	Natural evolutionary algorithm	<ul style="list-style-type: none"> Multi-objective VC allocation algorithm based on 4 different models: VM availability, power consumption, resource utilization, and load balance 	Cloud DC		CloudSim	<ul style="list-style-type: none"> Power consumption Average resources utilization, Risk value Load balance value 	Lower risk value, least power consumption, better average resources utilization with least load balance value
[32]	Dynamic VM placement	<ul style="list-style-type: none"> Cluster Management module Overload selection module Resource predictor module 	Cloud DC	<ul style="list-style-type: none"> Google Cluster Dataset 		<ul style="list-style-type: none"> Number of overloads Number of SLA-violation Power consumption 	Less number of overloads occurred with an outperforming on minimizing SLA violation.
[33]	Linear regression prediction model	<p>VM placement based on Linear regression prediction model: Overloaded and underloaded algorithms.</p> <ul style="list-style-type: none"> Live migration policies. 	Clouds DC	<ul style="list-style-type: none"> PlanetLab & rand workloads 	Simulation: CloudSim	<ul style="list-style-type: none"> Power consumption SLA violation 	Reduced power with minimum SLA violation.
[34]	Neural networks	Dynamic VM placement based on joint CPU demands & prediction of future VM placement.	Cloud	<ul style="list-style-type: none"> Real workloads 	Simulation: CloudSim	<ul style="list-style-type: none"> Energy consumption SLA violation 	Best result of reducing power consumption with smaller SLA violation.
[35]	Consider CPU and RAM as major power consumers	<ul style="list-style-type: none"> System consists of global and local manager. two algorithms for VM selection and placement with two thresholds for loading detections. 	CloudDC	<ul style="list-style-type: none"> Real dataset: SPECpower benchmark 	Simulation: CloudSim	<ul style="list-style-type: none"> Power consumption SLA violation Performance degradation due to migration 	Reduced power consumption and performance degradation with dynamic threshold.

Table 3. Cont.

Ref.	Major Feature	Main Tasks	Envi.	Dataset	Evaluation Tool	Measured Metrics	Contribution
[36]	Experimental evaluation of different policies, power management models, and power models.	Two algorithms which utilize FCFS/ SJF scheduling tasks and DVFC power management using cubic energy model.	Cloud DC	<ul style="list-style-type: none"> RIKEN Integrated Cluster of Clusters 	Simulation: CloudSim	<ul style="list-style-type: none"> Energy consumption 	Reduced power consumption.
[37]	Uninterruptable power supply UPS minimization cost function.	Centralized manager with two server-level agents: <ul style="list-style-type: none"> server power modular & local power controller. (PDM) 	Cloud	<ul style="list-style-type: none"> Computational-intensive application workload: mathematic prime counting functions. 	Testbed	<ul style="list-style-type: none"> Power (demand, assign, consume) CPU cap CPU frequency 	Power assignment applied properly which prevents UPSs overloading.
[38]	Three-phases for VM placement	Three-phases for VM management: <ul style="list-style-type: none"> Profiling & profile update Task classification & assignment Successive VM placement. 	Cloud DC	<ul style="list-style-type: none"> Google Cluster dataset 		<ul style="list-style-type: none"> VM number PH quantity gap Energy saving 	Energy saving is increased by at most 12%
[39]	Genetic Algorithm	<ul style="list-style-type: none"> VM selection manager Host selection maager 	SDN-Cloud	<ul style="list-style-type: none"> Synthetic and Wikipedia workload 	CloudSimSDN	<ul style="list-style-type: none"> Power consumption in Response time Traffic load, Hourly cost 	outperforms in reducing power consumption, response time, and hourly cost.

5. SDN-Based Fog Computing

Since cloud computing plays a major role in data storage, however, fog computing aims to reduce the latency between nodes, especially for delay-sensitive applications. Thus, one of the main reasons to develop SDN-based fog is to ease communication between nodes in order to ensure a reasonable performance. This section presents state-of-the-art researches that are conducted for SDN-fog computing to solve existing challenges specifically towards network performance issues.

An SDN-based solution is proposed in order to solve the complexity of data forwarding in smart grids in fog computing [40]. Three-tier architecture is proposed that consists of IoT as the bottom tier that is responsible on generating data, the middle tier, which is fog, enables smart grids that consist of smart meters (SM) and data collection servers (DCS). While the cloud tier, which is the top layer, consists of SDN controller and responsible in performing processing and permanent storage. Data forwarding uses shortest paths—specifically, Dijkstra Algorithm—with three types of path recovery that are forced when a link failure is encountered during data transmission. The proposed work is simulated in Mininet considering three types of topologies, which are large, medium and small to measure the transmission time, throughput and transmission overhead. The results show that the Dijkstra algorithm has the lowest forwarding time compared to other shortest path algorithms, i.e., Floyd-Warshall and Bellman-Ford. On the other hand, the forwarding time is increased when the path recovery conducted end-to-end link failure. However, no link failure case consumes more time than a path recovery conducted through switch-to-switch or switch-to-end cases when the topologies are large and medium, while the number of broken lines has a positive correlation with the forwarding time while a throughput has a negative correlation with the size of topology. While, in [41], multi-layer advanced networking environment middleware is proposed for the SDN-based fog environment to solve heterogeneity and different time requirements by different applications. Based on the application type, a forwarding mechanism is selected by the SDN controller such as by IP, payload content and overlay networks where selection and initiating are performed simultaneously. Real testbed is used for an intensive evaluation that considers mainly latencies and performances. The results show the capability of the proposed middleware to determine the best forwarding mechanism based on the application type and network status.

On the other hand, merging fog computing and SDN in the internet of vehicle is proposed to solve latency problem resulted from generated data by sensors [42]. It consists of five layers, including clusters of Internet of Vehicles (IoV), local and global load balancers, fog servers, SDN controller and cloud servers. Local and global load balancers are designed for local clusters by local load balancers and different clusters which will be performed by SDN controller. Considering CPU utilization as main criteria when determining size of load to be either overloaded or under loaded. A simulation was conducted via OMNET++ and SUMO with a size of vehicles equal to 1000 and evaluated against VANET-Cloud architecture. The results show that the proposed architecture and load balancers capable of increasing resource utilization, decreasing the average response time, outperforming in meeting tasks' deadlines, and uses lowest bandwidth compared to baseline.

Other research employed machine learning techniques to provide intelligence to SDN controllers, such in [43], a deep reinforcement learning was developed to perform task scheduling in the SDN controller by learning from intelligent agents that led to optimal decisions. The SDN controller is capable to generate and learn from historical datasets, filter and re-represent datasets, produce learning policies, selects nodes and activates rules. The evaluation is conducted via testbed using two datasets for training and testing. The results show that the maximum latency achieved equal to 12.5 milliseconds and saved energy up to 87%.

To solve the problem of packet delays imposed by the SDN controller in fog networking, a prediction model is proposed based on regression model which follow standard process of data mining methodology [44]. The process generally consists of data under-

standing, which aims to build an initial dataset, and data preparation, which aims to clean dataset, remove redundancy and construct features, as well, modeling, which includes selection and assessments machine learning algorithms that results in selecting random forest algorithm. The results shows that proposed model manages to predict delay values similar to actual values.

Moreover, a solution for SDN-fog services based on machine learning—specifically, reinforcement learning—is proposed in [45] in order to allocate service dynamically based on objectives such as power consumption, security and QoS constraints. Proposed reinforcement learning imposes recurrent neural networks in order to reduce the average response time. The service manager is built-in SDN controller specifies the locations of services to be processed dynamically by taking into consideration overall objective.

In [46], the proposed work goal is to increase storage, by reducing the cache redundancy and improving service placement with limited interactions with controller, therefore merging software-defined networks and named data network in fog computing is introduced. By extracting powerful features of both paradigms i.e., devoting controllability to SDN controller to perform routing, function placement, caching decision and service allocation. While the data plane is managed by forwarding information base, content store and pending interest table, which are offered by NDN stateful data plane. The evaluation is conducted using MATLAB against SDN-IP protocol systems by comparing the number of packets exchanged between the nodes and controller and number of new rules inserted by the controller. The results show SDN-NDN systems are capable of reducing communication between nodes and the controller.

In [47], data collection schema and estimation model for SDN-based vehicular fog computing is proposed. The architecture of SDN-based VFC consists of three layers, which are an infrastructure layer that includes vehicles, a fog layer that includes both regional controllers and local SDN controllers and a cloud layer, which includes a global SDN controller. The main tasks consist of the data collection configuration process, fog station deployments and data routing. A collection of data configuration consists of four main phases that are initialization of data collection parameters that are performed by a global SDN controller as data-oriented or fog-oriented. Then, the data collection parameter the sends collected data to fog stations. After that, users' requests are sent to fog stations. Finally, update data collection parameters to fit current environment, which include the deployment of fog station and data migration. A model is built to estimate number of fog stations required to be deployed and evaluation is conducted based on real traffic data. The results show that accurate estimation of fog stations is achieved, which depends on data availability and volume of data collection. While the data collection schema is evaluated in Veins Simulation Framework against classical schema to assess the latency, a number of hops to reach the fog station, recovery time and data loss. The results show that a proposed schema decreases the latency and number of hops. As well, less time is required for recovery, and less data loss is achieved.

The machine learning approach is utilized in [48], specifically the K-Nearest Neighbor (KNN) algorithm, together with multi-objective optimization—specifically, Non-dominated Sorting Genetic Algorithm (NSGA-II)—to meet the QoS constraint based on application type, either maximizing the path reliability or minimizing the delay. The system consists of three parts that are network architecture that consists of SDN switches, fog nodes, IoT devices and links associated with QoS parameters, i.e., bandwidth and transmission delay. Then, the communication model and provisioning model. In order to find the link reliability, the system uses machine learning techniques, i.e., KNN algorithm. While finding the optimal path as a tradeoff objective, the system uses the NSGA-II algorithm. An evaluation is conducted using Mininet with two types of applications: delay-sensitive that requires a path with less delay, and computational-intensive, which requires a reliable path. The results show that the system manages to meet each application QoS constraint by selecting required path according to optimal objective. Moreover, less delay and less packet loss are achieved compared to the baseline.

The software defined and multi-access edge computing vehicular network architecture is developed to reduce the latency in vehicular networks [49]. The system uses fuzzy logic for clustering vehicles and OpenFlow algorithm to update flow tables. The architecture consists of four layers, which are: access layer, that consists of forwarding devices. Then the forwarding layer, which consists of high-speed SDN switches. Then multi-edge computing layer that has local controllers that perform resources allocation, and the MEC application platform system, which performs cellular resource allocation, caching in-vehicle message modules, communication decision modules and multimedia broadcast multicast service computing modules. Finally, the control layer which contains global SDN controller. Using fuzzy logic for clustering the vehicles by head vehicle based on velocity, position and signal quality. After that, the SDN client switched flow tables are updated. The simulation was performed in OMNET++ with INET framework using real-time traffic via SUMO. The results show that proposed system manages to achieve the highest packet delivery ratio and lowest average E2E delay compared to the baselines.

Similarly, an offloading mechanism is proposed in [50] that works dynamically by considering nodes selection prior to offloading. The system is composed of fog nodes, which are connected to the SDN switch and whose information is gathered by fog agents. Additionally, core switches that contain several SDN switches that are controlled via SDN controller. As well, the controller that manages both fog nodes and network via the fog orchestration and SDN controller. Finally, the service layer, which includes fog support applications. On the other hand, a dynamic offloading service is triggered when it receives a call from fog orchestration, which is performed when a node sent an offload request to fog orchestration as a result of an overloading condition. However, the decision of which node will receive the offload that was is made after filtering the nominated nodes and ordered it based on the network status and available resources. The evaluation was conducted via the Minint emulation tool and Python, and iPref was used for traffic generation. Selection time, time to transfer data to the offloading node, throughput and available resources ratio are the evaluation metrics. The results showed that the proposed work was outperformed in minimizing the selection time, thus minimizing the delay. As well, it provides better throughput and less response time. As well, an artificial intelligent approach is proposed to overcome the delay and reliability issues in SDN-based IoV-Fog computing [51]. ARTNet architecture consists of the IoV layer which contains onboard devices, the fog layer, which contains base stations and RSU as the static models to provide processing and storage facilities. As well, the architecture contains an AI-based SDN that consists of three modules that are intelligent agents, deep learning and big data analytics. The experimental results show that the proposed algorithm outperforms in reducing the power consumption and latency with respect to the time slot and number of vehicles, especially when the number of IoV is increased. In addition to that, IoV architecture is proposed in [52], which consists of multiple fog clusters that contain several RSUs. RSUs gather information regarding roads and vehicles, then transform this information to RSU controller, which performs decision-making towards resource allocations in the fog layer. While the SDN controller resides in the cloud to obtain global view and forward status of the network with forwarding policies and rules to the RSU controller. Then, the resource allocation model is constructed, which considers several factors, which are the power consumption, delay, load balancing and time execution stability. After that, the proposed algorithm is developed with integration with the evolutionary algorithm by using hieratical clustering features to enhance the many-objective optimization algorithm. The experimental results show that the proposed algorithm outperforms in time execution stability and load balancing objectives.

Even though previous studies followed OpenFlow architecture, however, in [53], a multi-layer SDN Fog paradigm was proposed in order to organize the SDN controllers, locally and remotely, in fog computing using P4 and P4Runtime to overcome certain difficulties imposed by the OpenFlow protocol. Controllers can be built locally near edge devices in order to ease communication, while remote SDN controllers are centralized to control traffic, among other functions that require a global view of the network.

Table 5 summarizes the contributions related to SDN fog computing in terms of the major feature of algorithms, main tasks, the environments where the algorithm targets, which evaluation tool is used for testing and the dataset, evaluated metrics and, finally, the major contribution of the algorithm based on its published results. Figure 1 represents the relationship between each category.

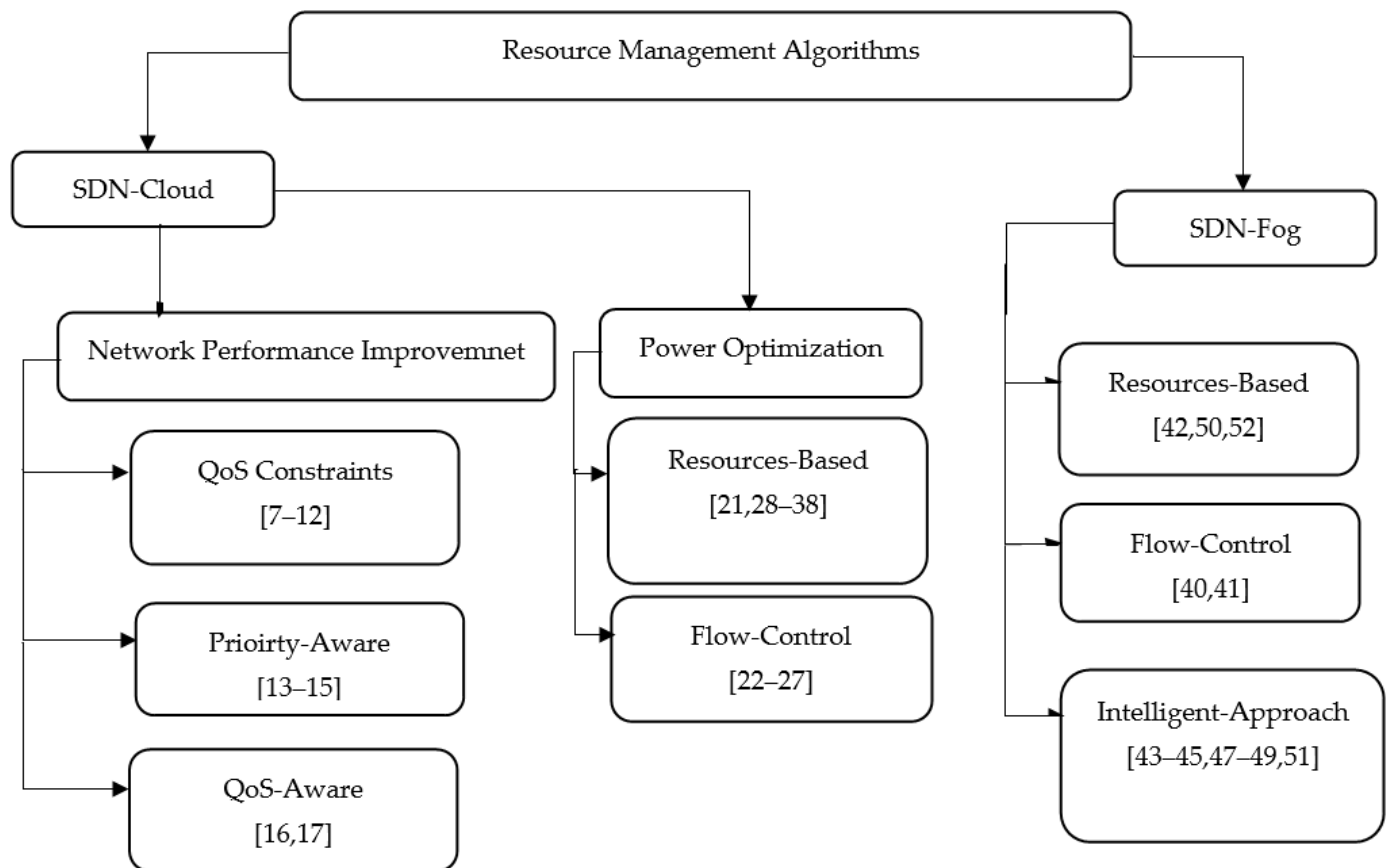


Figure 1. Taxonomy of reviewed research based on the main objective and research strategy.

Table 5. Summarization of the contributions in SDN-based fog computing.

Ref.	Major Feature	Main Tasks	Env.	Dataset	Eval.Tool	Measured Metrics	Contribution
[40]	<ul style="list-style-type: none"> data forwarding scheme 	<ul style="list-style-type: none"> data forwarding: Dijkstra algorithm 3 types path recovery 	sdn- fog enabled smart grid	<ul style="list-style-type: none"> 3 different sizes topologies 	mininet support openflow protocol	<ul style="list-style-type: none"> transmission time throughput transmission overhead 	<ul style="list-style-type: none"> Dijkstra has the lowest forwarding time compared to other shortest path algorithms Number of broken lines has a positive correlation with forwarding time. throughput has a negative correlation with the size of topology
[41]	<ul style="list-style-type: none"> Multi-layer routing approach 	<ul style="list-style-type: none"> Multi-layer advanced networking middleware which enables native IP, overlay networking information, and payload content data forwarding mechanisms. 	SDN- based Fog Computing	<ul style="list-style-type: none"> Varying size packets 	testbed	<ul style="list-style-type: none"> latency performance 	<ul style="list-style-type: none"> capability of determining best forwarding mechanism based on application type and network status.
[42]	Resource utilization using local and global load balancers	<ul style="list-style-type: none"> Five layers architecture Local and global load balancers based on CPU utilization and certain Thresholds 	SDN-Fog IoVs	<ul style="list-style-type: none"> 1000 vehicle with three clusters scenario 	OMNeT++ and SUMO	<ul style="list-style-type: none"> Resources utilization average response time percentage of meeting deadlines percentage of bandwidth usage 	<ul style="list-style-type: none"> increase resource utilization Decrease average response time Outperforms in meeting tasks' deadlines Uses lowest bandwidth
[43]	<ul style="list-style-type: none"> Delay prediction by Regression model 	<ul style="list-style-type: none"> three steps model: data understanding, preparation, and modeling. 	<ul style="list-style-type: none"> sdn-fogs 	<ul style="list-style-type: none"> uses several datasets with different number of instances. 	CRISP-DM methodology.	<ul style="list-style-type: none"> predicted and actual delay 	predicted delay values similar to actual delay values.

Table 5. Cont.

Ref.	Major Feature	Main Tasks	Env.	Dataset	Eval.Tool	Measured Metrics	Contribution
[44]	<ul style="list-style-type: none"> Deep reinforcement learning 	<ul style="list-style-type: none"> Tasks scheduler in SDN controller level 	<ul style="list-style-type: none"> SDN-based IoT 	<ul style="list-style-type: none"> Training set: 1000 episodes Evaluation set: 100 tasks 	Testbed	<ul style="list-style-type: none"> Saved energy Latency 	<ul style="list-style-type: none"> Reduced energy consumption Maximum latency equal to 12.5 milliseconds.
[45]	reinforcement learning	<ul style="list-style-type: none"> Service manager located to specify services locations based on function goal. 	<ul style="list-style-type: none"> SDN-Fog 	<ul style="list-style-type: none"> steady sequence of successive requests: 10 requests per second 	Testbed	<ul style="list-style-type: none"> Response time 	Ability to adopt to medium and high load.
[46]	<ul style="list-style-type: none"> Named Data Networking 	<ul style="list-style-type: none"> Jointing SDN-NDN: maintaining SDN controller capabilities and stateful dataplane in NDN. 	<ul style="list-style-type: none"> SDN-NDN Fog computing 	<ul style="list-style-type: none"> Varying number of services requests of size 1000 	MATLAB	<ul style="list-style-type: none"> number of exchange packets between controller and nodes Number of injected rules by controller. 	communication is reduced between nodes and controller
[47]	<ul style="list-style-type: none"> Data collection schema 	<ul style="list-style-type: none"> Three layers SDN-Vehicular Fog computing: infrastructure layer, fog layer, and cloud layer. 	<ul style="list-style-type: none"> SDN—based Vehicular Fog Computing. 	<ul style="list-style-type: none"> Real traffic measurements Different size clusters: 10–50 vehicles. 	Veins Simulation Framework	<ul style="list-style-type: none"> accurate model which estimates fog station based on data collection volume. reduced latency and number of hops reduced recovery time and data loss. 	

Table 5. Cont.

Ref.	Major Feature	Main Tasks	Env.	Dataset	Eval.Tool	Measured Metrics	Contribution
[48]	<ul style="list-style-type: none"> Machine learning: KNN algorithm multi-objective optimization algorithm: NSGA-II 	<ul style="list-style-type: none"> consists of network architecture, communication model, and provisioning model. 	<ul style="list-style-type: none"> SDN- enable IoT-Fog Computing 	<ul style="list-style-type: none"> training data: real life data recorded for time interval of 5 months testing data: real time and delay sensitive. testing data: computation-ally intensive tasks. 	Mininet with POX SDN controller	<ul style="list-style-type: none"> packet loss end to end delay 	paths are selected based on QoS constraint either reliable path/ minimize delay and succeeds on performing less data packets and delay.
[49]	<ul style="list-style-type: none"> Fuzzy logic 	4 layers architecture for SD-MEC: access layer, forwarding layer, multi edge computing layer, and control layer	<ul style="list-style-type: none"> SDN-based multi access edge for vehicle networks. 	<ul style="list-style-type: none"> Real world traffic conditions. 	OMNET++ with INET framework.	<ul style="list-style-type: none"> delay of in-vehicles messages end to end delay in respect of vehicle density. packet delivery ratio throughput 	least average E2E delay highest packet delivery ratio
[50]	<ul style="list-style-type: none"> two steps offloading- node selection 	4-layers architecture: fog layer, core switches, controller, and service layer	<ul style="list-style-type: none"> SDN-Fog 	<ul style="list-style-type: none"> iPref for traffic generation 	Minimit and Python	<ul style="list-style-type: none"> Selection time Time to transfer date to offloading node throughput Available resources ratio 	Outperforms in minmizing selection time thus minmizing delay. As well, provide better throughput and less response time.

Table 5. Cont.

Ref.	Major Feature	Main Tasks	Env.	Dataset	Eval.Tool	Measured Metrics	Contribution
[51]	<ul style="list-style-type: none"> Employing intelligent agents. 	3-layers architecture: IoV layer Fog layer Intellegent-SDN layer	<ul style="list-style-type: none"> SDN-IoV-Fog 		IoV based on SDV-F framework	<ul style="list-style-type: none"> energy consumption energy shortfall latency 	Least energy consumption and latency.
[52]	<ul style="list-style-type: none"> Two types of controllers are employed: RSUC and SDNC 	Multi- objective resource allocation	<ul style="list-style-type: none"> SDN-Fog-Cloud 			<ul style="list-style-type: none"> Stability Load balancing Power consumption Delay 	Outperform in time excution stability and load balancing objectives

6. Open Issues

Open issues related to SDN cloud and SDN fog, which are derived from a deep analysis of the above-mentioned reviewed researches, are listed as follows:

1. Power savings is a challenge, along with several related issues such as data transferring, powered devices, and energy wastage.
2. There is a great concentration on the limited number of metrics (Tables 3 and 4) when evaluating the network performance. On the other hand, power consumption is a major metric in power optimization algorithms, but other important measures of network performance are being neglected.
3. The network performance can be enhanced by considering the awareness of the resources management algorithm and considering the bandwidth due to its impact on the overall performance. Moreover, control over resources leads to performance improvement.
4. There has been always a tradeoff relationship between the network performance and power optimization due to the consequences resulting from overutilized and underutilized resources. Only few researches manage to find a balanced relationship between the two variables.
5. Dynamic deterministic of the overall objective from a set of objectives, such as the minimizing delay or maximizing reliability, based on the application type is an open issue.
6. Most research focused on the Central Processing Unit (CPU) as the most dominant factors in energy consumption. Even though it is considered as a high-power consumer element in PM, but other important factors are being neglected, such as RAM, network card, and storage.
7. Most presented techniques focused only at the server level and ignored the possible contributions on flows/links.
8. From state-of-the-art researches in SDN-based fog, we can observe that interrelated fields are presented such as machine learning and artificial intelligent in order to provide a fully programmed environment, which is a promising approach for future researches.
9. Identifying and fulfilling requests with higher priority prior to low-priority requests is an issue due to the high volume of data transferred among different devices, which results in congested links and, thus, latency.
10. Joining SDN and named data networking (NDN) is a new approach to gain the maximum privilege of both techniques. However, few researches are conducted in this field.
11. Furthermore, SDN cloud is an innovative field, which brings a great opportunity for researchers, since most studies either contribute on the field of cloud or SDN.
12. The scalability issue, which can be classified into two aspects: resources scalability either scaling up or down in the cloud and fog layers and controllers scalability (local and global) to consider their location and number of required controllers.
13. Security is one the main challenges encountered in SDN-Based cloud/fog, including authentication and SDN.
14. Legacy SDN such as OpenFlow forces certain architectures when designing controllers that lead to strict environment for the revolution and development, especially in a fog environment.
15. Only a few contributions regarding the simulation tools and testbed that facilitates the experiments of SDN-based environments.

7. Conclusions

From above discussion, we can conclude that SDN approach is a promising in the future to both clouds and fog computing due to its management simplicity advantages. This study overviewed novel and state-of-the-art findings in SDN cloud and SDN fog. Reviewed studies have been classified based on the main objective: network performance

improvement and energy optimization. Further classifications will be conducted based on algorithm strategies in achieving a desirable goal. The studies are summarized based on major feature of the proposed algorithm, main tasks, environment, evaluation tool, evaluation metric, dataset, and main contribution. After that, we overviewed related open issues, which covers several aspects, from the main objective to simulation tools development. SDN-based cloud/fog architecture plays an important role in scalability and performance. Moreover, merging SDN with other approaches such as artificial intelligent and machine learning to develop flexible and intelligent environments is promising in order to find an optimal solution to multi-objective algorithm and to achieve desirable outcomes in network performance and energy efficiency. As well, new contributions in designing SDN controller architecture other than OpenFlow are suggested.

Author Contributions: Author Contributions: Conceptualization, A.A. and S.K.S.; methodology, A.A. and S.K.S.; soft-ware, A.A.; validation, A.A.; formal analysis, A.A., S.K.S. and N.S.; investigation, A.A., S.K.S. and R.L.; resources, A.A., S.K.S. and Z.Z.; data curation, A.A.; writing—original draft prepara-tion, A.A.; writing—review and editing, S.K.S. and N.S.; visualization, A.A., S.K.S. and R.L.; su-pervision, S.K.S, N.S., R.L. and Z.Z.; project administration, S.K.S, N.S., R.L. and Z.Z.; funding acquisition, S.K.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was supported by the Universiti Putra Malaysia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Culver, P. *Software Defined Networks*, 2nd ed.; Morgan Kaufmann: Burlington, MA, USA, 2016.
2. Nunes, B.A.A.; Mendonca, M.; Nguyen, X.-N.; Obraczka, K.; Turletti, T. A survey of software-defined networking: Past, present, and future of programmable net-works. *IEEE Commun. Surv. Tutor* **2013**, *16*, 1617–1634. [\[CrossRef\]](#)
3. Leon-Garcia, A.; Bannazadeh, H.; Zhang, Q. Openflow and SDN for Clouds. *Cloud Serv. Netw. Manag.* **2015**, *6*, 129–152. [\[CrossRef\]](#)
4. Arivazhagan, C.; Natarajan, V. A Survey on Fog computing paradigms, Challenges and Opportunities in IoT. In Proceedings of the 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 28–30 July 2020; pp. 385–389.
5. Schaller, S.; Hood, D. Software defined networking architecture standardization. *Comput. Stand. Interfaces* **2017**, *54*, 197–202. [\[CrossRef\]](#)
6. Gonzalez, N.M.; Carvalho, T.C.M.D.B.; Miers, C.C. Cloud resource management: Towards efficient execution of large-scale scientific applications and workflows on complex infrastructures. *J. Cloud Comput.* **2017**, *6*, 13. [\[CrossRef\]](#)
7. Wang, R.; Esteves, R.; Shi, L.; Wickboldt, J.A.; Jennings, B.; Granville, L.Z. Network-aware placement of virtual machine ensembles using effective bandwidth estimation. In Proceedings of the 10th International Conference on Network and Service Management (CNSM) and Workshop, Rio de Janeiro, Brazil, 17–21 November 2014; pp. 100–108.
8. Ballani, H.; Costa, P.; Karagiannis, T.; Rowstron, A. Towards predictable datacenter networks. *ACM SIGCOMM Comput. Commun. Rev.* **2011**, *41*, 242–253. [\[CrossRef\]](#)
9. Wang, R.; Wickboldt, J.A.; Esteves, R.P.; Shi, L.; Jennings, B.; Granville, L.Z. Using empirical estimates of effective band-width in network-aware placement of virtual machines in datacenters. *IEEE Trans. Netw. Serv. Manag.* **2016**, *13*, 267–280. [\[CrossRef\]](#)
10. Wang, R.; Mangiante, S.; Davy, A.; Shi, L.; Jennings, B. QoS-aware multipathing in datacenters using effective bandwidth estimation and SDN. In Proceedings of the 2016 12th International Conference on Network and Service Management (CNSM), Montreal, QC, Canada, 31 October–4 November 2016; pp. 342–347.
11. Hopps, C. “Analysis of an Equal-Cost Multi-Path Algorithm,” Internet Requests for Comments, RFC 2992. November 2000. Available online: <https://www.hjp.at/doc/rfc/rfc2992.html> (accessed on 5 March 2021).
12. Al-Mansoori, A.; Abawajy, J.; Chowdhury, M. BDSP in the cloud: Scheduling and Load Balancing utilizing SDN and CEP. In Proceedings of the 2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID), Melbourne, VIC, Australia, 11–14 May 2020; pp. 827–835.
13. Tsai, W.-T.; Shao, Q.; Elston, J. Prioritizing Service Requests on Cloud with Multi-tenancy. In Proceedings of the 2010 IEEE 7th International Conference on E-Business Engineering, Shanghai, China, 10–12 November 2010; pp. 117–124.
14. Ellens, W.; Ivkovic, M.; Akkerboom, J.; Litjens, R.; van den Berg, H. Performance of cloud computing centers with multi-ple priority classes. In Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing, Honolulu, HI, USA, 24–29 June 2012; pp. 245–252.

15. Son, J.; Buyya, R. Priority-Aware VM Allocation and Network Bandwidth Provisioning in Software-Defined Networking (SDN)-Enabled Clouds. *IEEE Trans. Sustain. Comput.* **2018**, *4*, 17–28. [[CrossRef](#)]
16. Lin, S.-C.; Wang, P.; Luo, M. Jointly optimized QoS-aware virtualization and routing in software defined networks. *Comput. Netw.* **2016**, *96*, 69–78. [[CrossRef](#)]
17. Wang, S.-H.; Huang, P.P.W.; Wen, C.H.P.; Wang, L.C. EQVMP: Energy-efficient and qos-aware virtual machine placement for software defined datacenter networks. In Proceedings of the International Conference on Information Networking 2014 (ICOIN2014), Phuket, Thailand, 10–12 February 2014; pp. 220–225.
18. Cziva, R.; Jout, S.; Stapleton, D.; Tso, F.P.; Pezaros, D.P. SDN-based virtual machine management for cloud data centers. *IEEE Trans. Netw. Serv. Manag.* **2016**, *13*, 212–225. [[CrossRef](#)]
19. Tso, F.P.; Oikonomou, K.; Kavvadia, E.; Pezaros, D.P.; Oikonomou, K. Scalable Traffic-Aware Virtual Machine Management for Cloud Data Centers. In Proceedings of the 2014 IEEE 34th International Conference on Distributed Computing Systems, Madrid, Spain, 30 June–3 July 2014; pp. 238–247.
20. Mann, V.; Gupta, A.; Dutta, P.; Vishnoi, A.; Bhattacharya, P.; Poddar, R.; Iyer, A. Remedy: Network-aware steady state VM management for data centers. In Proceedings of the International Conference on Research in Networking, Prague, Czech Republic, 21–25 May 2012; Volume 7289, pp. 190–204.
21. Son, J.; Dastjerdi, A.V.; Calheiros, R.N.; Buyya, R. SLA-Aware and Energy-Efficient Dynamic Overbooking in SDN-Based Cloud Data Centers. *IEEE Trans. Sustain. Comput.* **2017**, *2*, 76–89. [[CrossRef](#)]
22. Zheng, K.; Wang, X. Dynamic Control of Flow Completion Time for Power Efficiency of Data Center Networks. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 340–350.
23. Zheng, K.; Wang, X.; Wang, X. PowerFCT: Power optimization of data center network with flow completion time constraints, IPDPS. In Proceedings of the 2015 IEEE International Parallel and Distributed Processing Symposium, Hyderabad, India, 25–29 May 2015.
24. Zheng, K.; Wang, X.; Liu, J. DISCO: Distributed traffic flow consolidation for power efficient data center network. In Proceedings of the 2017 IFIP Networking Conference (IFIP Networking) and Workshops, Stockholm, Sweden, 12–16 June 2017; pp. 1–9.
25. Heller, B.; Seetharaman, S.; Mahadevan, P.; Yiakoumis, Y.; Sharma, P.; Banerjee, S.; McKeown, N. ElasticTree: Saving energy in data center networks. In Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2010), San Jose, CA, USA, 28–30 April 2010; p. 17.
26. Wang, X.; Yao, Y.; Wang, X.; Lu, K.; Cao, Q. CARPO: Correlation-aware power optimization in data center networks. In Proceedings of the 2012 Proceedings IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012; pp. 1125–1133.
27. Alizadeh, M.; Greenberg, A.; Maltz, D.A.; Padhye, J.; Patel, P.; Prabhakar, B.; Sengupta, S.; Sridharan, M. Data center TCP (DCTCP). In Proceedings of the SIGCOMM '10: Proceedings of the ACM SIGCOMM 2010 Conference, New York, NY, USA, 30 August–3 September 2010; pp. 63–74.
28. Abts, D.; Marty, M.R.; Wells, P.M.; Klausler, P.; Liu, H. Energy proportional datacenter networks. *ACM SIGARCH Comput. Arch. News* **2010**, *38*, 338–347. [[CrossRef](#)]
29. Zhao, J.Z.; Li, K. An Energy-Aware Algorithm for Virtual Machine Placement in Cloud Computing. *IEEE Access* **2019**, *7*, 55659–55668. [[CrossRef](#)]
30. Sharma, N.K.; Reddy, G.R.M. Multi-Objective Energy Efficient Virtual Machines Allocation at the Cloud Data Center. *IEEE Trans. Serv. Comput.* **2019**, *12*, 158–171. [[CrossRef](#)]
31. Liu, X.; Cheng, B.; Wang, S. Availability-Aware and Energy-Efficient Virtual Cluster Allocation Based on Multi-Objective Optimization in Cloud Datacenters. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 972–985. [[CrossRef](#)]
32. Alanazi, S.; Hamdaoui, B. Energy-Aware Resource Management Framework for Overbooked Cloud Data Centers with SLA Assurance. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6.
33. Li, L.; Dong, J.; Zuo, D.; Wu, J. SLA-Aware and Energy-Efficient VM Consolidation in Cloud Data Centers Using Robust Linear Regression Prediction Model. *IEEE Access* **2019**, *7*, 9490–9500. [[CrossRef](#)]
34. Shaw, R.; Howley, E.; Barrett, E.A. Predictive Anti-Correlated Virtual Machine Placement Algorithm for Green Cloud Computing. In Proceedings of the 2018 IEEE/ACM 11th International Conference on Utility and Cloud Computing (UCC), Zurich, Switzerland, 17–20 December 2018; pp. 267–276.
35. Gul, B.; Khan, I.A.; Mustafa, S.; Khalid, O.; Hussain, S.S.; Dancey, D.; Nawaz, R. CPU and RAM Energy-Based SLA-Aware Workload Consolidation Techniques for Clouds. *IEEE Access* **2020**, *8*, 62990–63003. [[CrossRef](#)]
36. Than, M.M.; Thein, T. Energy-Saving Resource Allocation in Cloud Data Centers. In Proceedings of the 2020 IEEE Conference on Computer Applications (ICCA), Yangon, Myanmar, 27–28 February 2020; pp. 1–6.
37. Al-Hazemi, F.; Lorincz, J.; Mohammed, A.F.Y. Minimizing Data Center Uninterruptable Power Supply Overload by Server Power Capping. *IEEE Commun. Lett.* **2019**, *23*, 1342–1346. [[CrossRef](#)]
38. Ding, Z.; Tian, Y.; Tang, M.; Li, Y.; Wang, Y.; Zhou, C. Profile-Guided Three-Phase Virtual Resource Management for Energy Efficiency of Data Centers. *IEEE Trans. Ind. Electron.* **2020**, *67*, 2460–2468. [[CrossRef](#)]
39. Rawas, S. Energy, network, and application-aware virtual machine placement model in SDN-enabled large scale cloud data centers. *Multimed. Tools Appl.* **2021**, 1–22. [[CrossRef](#)]

40. Okay, F.Y.; Ozdemir, S.; Demirci, M. SDN-Based Data Forwarding in Fog-Enabled Smart Grids. In Proceedings of the 2019 1st Global Power Energy and Communication Conference (GPECOM), Nevsehir, Turkey, 12–15 June 2019; pp. 62–67.
41. Bellavista, P.; Giannelli, C.; Montenero, D.D.P. A Reference Model and Prototype Implementation for SDN-Based Multi Layer Routing in Fog Environments. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 1460–1473. [[CrossRef](#)]
42. Kadhim, A.J.; Seno, S.A.H. Maximizing the Utilization of Fog Computing in Internet of Vehicle Using SDN. *IEEE Commun. Lett.* **2018**, *23*, 140–143. [[CrossRef](#)]
43. Sellami, B.; Hakiri, A.; Ben Yahia, S.; Berthou, P. Deep Reinforcement Learning for Energy-Efficient Task Scheduling in SDN-based IoT Network. In Proceedings of the 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 24–27 November 2020; pp. 1–4.
44. Casas-Velasco, D.M.; Villota-Jacome, W.F.; Da Fonseca, N.L.S.; Rendon, O.M.C. Delay Estimation in Fogs Based on Software-Defined Networking. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
45. Frohlich, P.; Gelenbe, E.; Nowak, M.P. Smart SDN Management of Fog Services. In Proceedings of the 2020 Global Internet of Things Summit (GloTS), Dublin, Ireland, 3 June 2020; pp. 1–6.
46. Amadeo, M.; Campolo, C.; Ruggeri, G.; Molinaro, A.; Iera, A. Towards Software-defined Fog Computing via Named Data Networking. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 29 April–2 May 2019; pp. 133–138.
47. Boualouache, A.; Soua, R.; Engel, T. Toward an SDN-based Data Collection Scheme for Vehicular Fog Computing. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6.
48. Akbar, A.; Ibrar, M.; Jan, M.A.; Bashir, A.K.; Wang, L. SDN-Enabled Adaptive and Reliable Communication in IoT-Fog Environment Using Machine Learning and Multiobjective Optimization. *IEEE Internet Things J.* **2021**, *8*, 3057–3065. [[CrossRef](#)]
49. Nkenyereye, L.; Nkenyereye, L.; Islam, S.M.R.; Kerrache, C.A.; Abdullah-Al-Wadud, M.; Alamri, A. Software Defined Network-Based Multi-Access Edge Framework for Vehicular Networks. *IEEE Access* **2019**, *8*, 4220–4234. [[CrossRef](#)]
50. Phan, L.-A.; Nguyen, D.-T.; Lee, M.; Park, D.-H.; Kim, T. Dynamic fog-to-fog offloading in SDN-based fog computing systems. *Future Gener. Comput. Syst.* **2021**, *117*, 486–497. [[CrossRef](#)]
51. Ibrar, M.; Akbar, A.; Jan, R.; Jan, M.A.; Wang, L.; Song, H.; Shah, N. ARTNet: AI-based Resource Allocation and Task Offloading in a Reconfigurable Internet of Vehicular Networks. *IEEE Trans. Netw. Sci. Eng.* **2020**, *1*. [[CrossRef](#)]
52. Cao, B.; Sun, Z.; Zhang, J.; Gu, Y. Resource Allocation in 5G IoV Architecture Based on SDN and Fog-Cloud Computing. *IEEE Trans. Intell. Transp. Syst.* **2021**, 1–9. [[CrossRef](#)]
53. Ollora Zaballa, E.; Franco, D.; Aguado, M.; Berger, M.S. Next-generation SDN and fog computing: A new paradigm for SDN-based edge computing. In Proceedings of the 2nd Workshop on Fog Computing and the IoT, Sydney, Australia, 8 April 2020; pp. 9:1–9:8.