Review of Path Selection Algorithms with Link Quality and Critical Switch Aware for Heterogeneous Traffic in SDN

Review Paper

Muhammad Nura Yusuf
Faculty of Computing, Univerisiti Teknologi Malaysia, Johor 81310, Malaysia and
Department of Mathematical Science, Abubakar Tafawa Balewa University, Bauchi PMB 0284, Nigeria
ymnura@atbu.edu.ng

Kamalrulnizam bin Abu Bakar
Faculty of Computing, Univerisiti Teknologi Malaysia, Johor 81310, Malaysia
knizam@utm.my

Babangida Isyaku
Sule Lamido University, PMB 047 Kafin Hausa, Jigawa State, Nigeria
Babangida.isyaku@slu.edu.ng

Ajibade Lukuman Saheed
Faculty of Computing, Univerisiti Teknologi Malaysia, Johor 81310, Malaysia
saheed2066@graduate.utm.my

Abstract – *Software Defined Networking (SDN) introduced network management flexibility that eludes traditional network architecture. Nevertheless, the pervasive demand for various cloud computing services with different levels of Quality of Service requirements in our contemporary world made network service provisioning challenging. One of these challenges is path selection (PS) for routing heterogeneous traffic with end-to-end quality of service support specific to each traffic class. The challenge had gotten the research community's attention to the extent that many PSAs were proposed. However, a gap still exists that calls for further study. This paper reviews the existing PSA and the Baseline Shortest Path Algorithms (BSPA) upon which many relevant PSA(s) are built to help identify these gaps. The paper categorizes the PSAs into four, based on their path selection criteria, (1) PSAs that use static or dynamic link quality to guide PSD, (2) PSAs that consider the criticality of switch in terms of an update operation, FlowTable limitation or port capacity to guide PSD, (3) PSAs that consider flow variabilities to guide PSD and (4) The PSAs that use ML optimization in their PSD. We then reviewed and compared the techniques' design in each category against the identified SDN PSA design objectives, solution approach, BSPA, and validation approaches. Finally, the paper recommends directions for further research.*

Keywords: *Software Defined Networking (SDN), Path Selection Algorithms (PSA), Routing, Quality of Service (QoS), Traffic Management, Flow table Management*

1. INTRODUCTION

The proliferation of the Internet of Things (IoT) applications has significantly increased the number and heterogeneity of traffic in modern networks[1]. A prior study reveals that active internet devices will rise from 26.66 billion in 2019 to 41 billion in 2027[3]. The signs of these become apparent during the COVID-19 pandemic [2]. The traffic arrival rate will increase the traffic volume on the internet and the rate of new path setup requests. Furthermore, the traffic flow is heterogeneous because they show non-uniform arrival rate, duration, and size. The heterogeneity affects their quality of service (QoS) requirements and demands of network resources. They behave differently en route to their destination.

Large flows like Elephant Flows (EF) are very few, about 1 -10 % of total network traffic. However, they are Long-Lived (LLF) and tend to consume network buffers. Their behaviour consequently imposes congestion and delays to most Mice Flows(MF)[4]. For instance, applications like Hadoop demand an enormous amount of throughput to perform an all-to-all transfer of petabytes during the shuffle phase. This demand is similar to a virtual machine migration that consumes high bandwidth. On the other hand, MFs are delay-sensitive and require high-priority queues. These technologies exhibit characteristic flows that require a highly dynamic network technology to meet their requirements[5].

However, techniques like Equal Cost Multiple Paths (ECMP) [6] do not distinguish among flows during routing. It amalgamates flows, irrespective of their requirements, on the same path. Using ECMP can lead to switching buffer overflows and inefficient bandwidth utilization. Meanwhile, critical traffic must arrive at their destination free of any delays. In contrast, bandwidth-intensive traffics must be assured a high throughput. For these reasons, efficient resource management and optimum path selection specific to each traffic are crucial to the operation of modern Data Centre Networks (DCN)[7], Wide Area Networks (WAN)[8], Enterprise Networks [9], and Internet Exchange Point Networks (IXP)[10]. This paper studies networks management applications' ability to adapt to the needs of these technologies.

Routing involves designing network policies and configuring devices to send a flow from source to destination. However, a traditional network [11] is unsuitable for this task. The unsuitability is because the devices in traditional are distributed and do not have a global knowledge of the entire network. As such, the network operator must be on-site to adhere to vendor specifications during configuration [12–14]. Thus, network management is not flexible and precise [15]. Meanwhile, the emergence of Software Defined Networking (SDN) [16] provided a new paradigm with better flexibility to efficiently manage a network in a way that overcomes most of the limitations of traditional architecture. This paper explores how SDN architecture handles path selection for traffic routing.

The SDN's flexibility comes from separating the Control Plane (CP) from the Data Plane (DP). With the separation, all network control functions are programmed centrally as opposed to the traditional network. The centralization frees the DP to focus on forwarding packets only. The controller abstracts the DP from all network applications at Application Plane (AP) and communicates only through Northbound (NBi) and Southbound (SBi) Interfaces. The controller extracts the network statistics from the DP in real-time to feed various network applications. Examples of these applications are routing [1], security [17], congestion [18], prioritized services [19], QoS [20], load balancing [21], energy [22], and many others. The applications run their algorithms to implement new rules simultaneously throughout the network seamlessly. So, the controller can dynamically make changes to a network in response to events like the arrival of a new flow, traffic bursts, or topology changes [23]. Other reasons could be intrusion detection, a node, or link failure, which may occur every 30min in large networks [2]. In any of these situations, a route computation process to select another path is triggered to update the DP with new rules. The update operation is done proactively using protection or reactively using restoration approaches [24]. In both cases, the key challenge for the controller and the underlying Path Selection Algorithms (PSA) is to swiftly complete the DP's operations swiftly with the least convergence time and transient congestion.

The frequency of rules update is 1.5 to 5s in the restoration approach[1]. The time spent to complete one cycle of update operation consists of (1) the rule computation delay, (2) the transmission and propagation delay in distributing the rule to all the switches, and (3) the delay in installing the rule. Depending on the number of switches along the path, the process can increase the network convergence time. It is challenging for the controller to complete this operation within the carrier-grade network requirement of 50ms[24] or 10ms when dealing with critical data [25]. As reported by [26,27], the operation might be even more complex within a few milliseconds requirement of a vehicle communication system and IoT applications. Another factor to consider is the differences in controllers' processing abilities. Maestaro[28], and NOX[29] can handle 600,000 and 30,000 flow requests/s, respectively, but Ryu can only support 6000 flow requests/s [30].

Although the aggressive use of wildcards[31–36] in the protection approach may help reduce the communication overhead and yield faster forwarding performance. However, TCAM's limitation to supporting only 2000 rule entries makes this approach less attractive [3]. Thus, the restoration approach is receiving significant attention because of its adaptability to contemporary network dynamics [1]. Although at the detriment of communication overhead and the challenge of strict adherence to the conflicting QoS requirements. Given the importance of the field, various techniques have been proposed lately. However, flexibility in customization to adapt to traffic variabilities in modern-day networks is still limited. This limitation calls for exploring and reviewing more path selection criteria to provide researchers with valuable references in state-of-the-art to do more work in the field. This paper critically reviews the proposed techniques within Ten (10) years. Other studies have conducted a similar study.

For instance, Khan et al. in [5] surveyed QoS provision techniques in Service-Oriented Architecture SOA on SDN. Segment, static, and dynamic link cost routing based on Fog-Enabled IoT Platform, have been surveyed in [23,37,[38] respectively. On the contrary, this paper focused on SDN-PSA in various use cases in WAN, DCN, and WSN. Karakus and Duresi [39] partially survey routing for multimedia flows. In contrast, this paper focused on SDN PSA, classified based on path selection criteria. Tomovic et al. [40] and Guck [41] compared the QoS routing in large-scale SDN with a focus on bandwidth and delay. Guck's study focused on unicast communication and considered only the algorithms that find a single path. This paper compares general communication techniques (multicast) and multipath algorithms. Likewise, low latency transmission strategies in SDN have been surveyed in [25]. In contrast, this paper covers other QoS metrics, such as Delay, Loss, and Bandwidth during path selection. Load balancing has been covered in [15], [42–44]. The papers discussed approaches such as Controller Placement Problem (CPP) [45] and Switch Migration (SM) [46]. On the other hand, this paper focuses on PSA techniques at csCP and dmCP for load balancing, along with QoS and practical resource usage. Waziral et al. survey issues related to topology discovery in [47]. SDN's energy issues were surveyed in [49–51]. On the contrary, this paper covers other PS goals, like Load Balancing and Resource Utilization, using various methods other than ML. This paper identifies the baseline algorithm used for PS and classifies the PS based on Selection Criteria (PSC). Then compares them for use cases, design goals, selection constraints, solution approach, and validation techniques. Table 1 provides a comparison summary to highlight these differences.

The key contributions of this research are as follows:

- The paper identified the baseline shortest path (sp) algorithms used for path selection in SDN.
- The paper Identifies different PSA problem types and design objectives in SDN
- Provide a classification of PSA based on Link Quality, Switch role, and Flow characteristics
- Finally, the study identifies and discusses potential future research directions

Figure 1 provides the paper's organizational chart. Section 2.gives an overview of Path Selection Algorithms in SDN. The overview covers how a PSA works in SDN, highlighting some baseline algorithms upon which many PSAs are built. The section also provides design objectives and different PSA problem types. It concludes by introducing the PSA classifications. Section 3. provided the critical review in four (4) sub-sections according to Link Quality, Switch role, Flow characteristics, and Machine learning approaches. Section 4 provides future research directions. Finally, section 5. concludes the paper.

Table 1: Comparison of Related Papers: **RP**: Routing Path, **SA**: Solution Approaches, **PSC**: Path Selection Criteria, **PSCr**: Path Selection Constraints, **ML**: Machine Learning, **IET**: Implementation and Evaluation Tools, **LB**: Load Balance, **RU**: Resource Utilization, **FT**: Fault tolerance

Ref.	Comm Tech		Routing Path		Use Case			Design Goal (DG)				PS Techniques Classification				IET	Missing aspect in contrast to this paper
	Uni cast	Multi cast	Single	Multi	Wired	Wire less	IoT	LB	QoS	RU	FT	SA & Problem Formulation		PSCr	PSC		
												ML	Others				
[5]	NA		X	X	SOA			X	✓	✓	X	X	✓	X	X	✓	Comm Tech, FT, ML, PSCr, PSC,
[15]	NA		✓	✓	Both		X	✓	X	X	X	X	✓	X	X	X	Comm Tech, QoS, RU, FT, ML, PSCr, PSC, IEA
[23]	✓	X	✓		L2&3VPN, DCN			X	✓	✓	✓	X	Optimization	X	✓	✓	Multi cast, Multi path, LB, ML, PSCr
[25]	X	X	Not Stated		Not Identified			✓	✓	✓	X	✓	✓	X	X	✓	Comm Tech, Use Case, PSCr, PSC,
[37]	✓	X	Single		Wired			X	✓	X	X	X	Optimization	✓	✓	✓	Comm Tech, Multi-Path, IoT, LB, RU, FT,
[38]	✓	X	Both		Fog, IoT			✓	✓	X	X	X	✓	X	X	✓	Multi cast, RU, FT
[39]	NA		Not Stated		WAN			X	✓	X	X	X	Optimization	X	X	X	Comm Tech, PSCr, PSC Path, IEA
[40]	X	X	Not Stated		WAN			X	✓	X	X	X	✓	✓	✓	X	Comm Tech, Path, IEA
[41]	✓	X	✓	X	✓	X	X	X	✓	X	X	X	✓	✓	✓	X	IEA, LB, RU
[42]	X	X	X	X	Both		X	✓	X	X	X	X	✓	X	X	X	Comm Tech, QoS, RU, ML, PSCr, PSC,
[43]	X	X	X	X	Both		X	✓	X	X	X	X	✓	X	X	X	Comm Tech, QoS, RU, ML, PSCr, PSC, IEA
[44]	Not Specified		X	X	cSC, dDC, & 5G			✓	✓	X	X	✓	✓	X	X	X	Comm Tech, RP, RU, FT, PSCr, PSC, IEA
[48]	X	X	X	X	✓	✓	X	X	✓	✓	X	✓	X	X	X	✓	Comm Tech, Path, Env, SA,
[49]	NA		X	X	Energy, DCN			X	✓	✓	X	X	Optimization	X	X	✓	Comm Tech, RP, LB, QoS, ML, PSCr, PSC
[50]	NA		Not Stated		Energy			✓	X	✓	X	X	Optimization	X	✓	X	Comm Tech, RP, QoS, FT, ML, PSCr&IET
[51]	NA		Sleep Schedule		Wired		X	X	✓	✓	X	X	Optimization	X	X	X	Comm Tech, LB, FT, PSCr, PSC, IET
This Paper	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

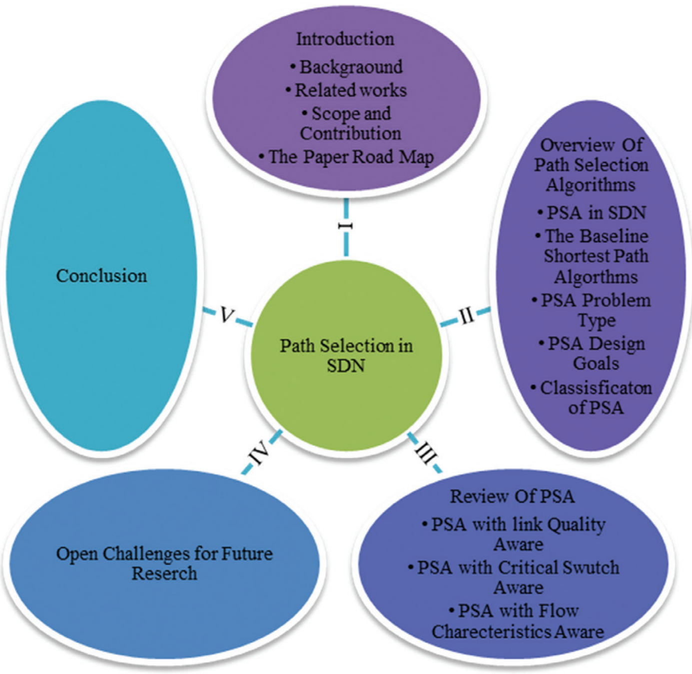

Fig. 1. Organization of the Survey

2. OVERVIEW OF SDN PATH SELECTION ALGORITHMS

2.1. PATH SELECTION ALGORITHM (PSA) IN SDN

SDN controller used an OpenFlow [52] to communicate with the DP. The standard uses a Link Layer Discovery Protocol (LLDP) to discover the DP topology. Upon discovery, the controller uses a PSA to compute a rule that guides flow to their destinations. It then instructs the switches to install the rules in their respective flow tables. In most cases, a PSA is invoked whenever a new flow with no corresponding entry in the flow table arrives. Another reason could arise from topology changes due to link or node failure. In both situations, the PSA must converge the network with the new rule to avoid disruptions. The controller uses a thread monitoring mechanism to track these changes by collecting statistics in a fixed cycle processing. The mechanism periodically issues a request to switches to get the information. In the Ryu framework [53], the issuance of information requests for all the registered switches is repeated infinitely after a set time interval. The network rules must be regularly updated every 1.5 to 5s due to traffic variabilities [1].

2.2. BASELINE SHORTEST PATH (SP) ALGORITHMS USED FOR PS IN SDN

Most of the PSA for SDN often use mechanisms of the Shortest Path problem as a foundation. Some of these are based on algorithms like Dijkstra[54], A* [55], or Bellman-Ford[56]. Others are based on k Shortest Paths (kSP), like Yen's algorithm [57]. In contrast, others use Restricted Shortest Path (kRSP) or Constrained Shortest Path (kCSP) like A* Prune [58]. These algorithms are proven to be NP-Complete [59]. So some exact algorithms for handling the problem in a brute-force manner or approximation like Lagrangian Relaxation Based Aggregated Cost (LARAC) are used.

Dijkstra [54] is a centralized algorithm that calculates the Shortest Path (SP) to multiple destinations in a graph with non-negative link weight. It keeps a queue with a list of partial paths starting at the source and going through intermediate nodes before the destination. The least weighted path from the queue is chosen at each iteration. So n paths are created by extending the partial paths to n external links of the node where the path ends. Only routes with a lower weight than the queue's current route that led to the same destination are enqueued. Depending on the weight values, the Dijkstra loosens up by removing a path with a higher weight value. The algorithm uses a Breadth-First Search (BFS) to visit a node.

On the other hand, the **A* algorithm** [55] is an enhancement of [54] by introducing a guess function at each node to estimate the path's cost from the reference node to the terminal node. The priority queue's outlier paths with minimum projected cost are expanded initially. The nearer the estimated cost is to the actual cost, the faster the algorithm converges. However, the overhead brought forth by the computation of the guess function constitutes the trade-off to consider. Furthermore, in contrast to [54], a **Bellman-Ford** (BFA) [56] computes the Shortest Path (SP) tree in a network with negative edge weight. Unlike Dijkstra, BFA operates in a distributed manner. The algorithm keeps track of the best path connected to V nodes. It performs |V|-1 number of iterations to update each node's current best path. This way, all SPs will ultimately be found because the path to any node is only|V|-1. It instantly halted if an iteration yielded no update because successive iterations would not result in any change.

The earliest algorithm to address a k-shortest path problem is **Yen's** [57]. The algorithm runs in two (2) phases. In the first phase, it works based on traditional SPA to find the initial path. Then, the other (k-1) shortest paths are found regarding the initial path's intermediate switches. The algorithm is similar to [55] and always stops once the terminal switch is visited k times. Additionally, **the A* prune** [58] identifies k-SP from source to destination in a communication network. The technique is an exact algorithm similar to [55] in using a guess function for each metric. Paths are considered by adjusting the cost estimation and pruning according to the cost projection exceeding the corresponding end-to-end bound. The process continues to iterate until k-CSPs are found or the candidate path container is empty.

To deal with NP-Complete problems such as kRSP or kCSP, a Lagrangian Relaxation Based Aggregated Cost (LARAC) [62] is used to relax the CSP with a combined delay and link cost to SP problem with a modified cost function. The method allows dropping some constraints of the first problem and introducing them in the optimization goal. However, the technique has a duality gap deficiency because it does not guarantee the best path return.

2.3. THE DESIGN GOAL/OBJECTIVES OF PSA

In many cases, PSA algorithms' design goal is threefold: QoS satisfaction, Resource Utilization, and Load Balancing.

QoS satisfaction deals with the ability of a network to consistently adhere to the performance expected by an application in terms of delay, jitter, bandwidth, throughput, and loss. QoS routing is one of the most critical components of a network management framework [41]. Ensuring it in a path is challenging due to network dynamics. Users' conflicting interests compound the difficulty, as exemplified by the high demands of pervasive applications like VoIP, video conferencing, telemedicine, and online game. Many routing strategies are designed to adapt to these demands and select a route with optimized QoS requirements [5]. Other PSAs are designed with resource utilization objectives. These PSA allocate network resources such as CPU, Memory, and bandwidth to traffic based on availability, priority, and requirement [61–63].

Lastly, other PSAs are designed to distribute network loads for task processing, packet transmission, and storage to network components based on their residual capacity [42]. The default settings of control algorithms are often similar across all CP[25]. In most cases, they are based on Shortest Path Algorithms (SPA) such as [54],56]. For example, the default setting of the Beacon [64] is [54]. The situation conditioned the CP to take the same PSD, irrespective of network conditions. However, some paths are more appealing than others, thus becoming critical as all the switches often select a node connected to them as the next hop. Thus, many flows are sent to the same path simultaneously. However, suppose the network experiences a traffic burst caused by some hot events. In that case, the component already running at full will be under added strain. The unbalanced distribution of the traffic along paths can lead to some paths becoming congested. Consequently, the network begins to experience delay and packet loss, resulting in an ultimate failure if the situation persists. Some PSAs are designed with load-balancing objectives to mitigate this situation at th DP. Another thing to note is that load imbalance affects the CP in response time delay as it affects the DP[25]. A dmCP can be overwhelmed due to an imbalanced distribution of flow request processing tasks. At the CP, the problem is being addressed through Controller Placement Problem (CPP) solutions [65–67] and Switch Migration (SM) [68–72]. Therefore, the load balance problem is addressed through the PSA at the DP and CPP or SM at CP. This paper focused on the PSA for load balancing at the DP.

2.4. PSA PROBLEM TYPE

A PSA is designed to best-effort traffic or for traffic with stringent QoS requirements. Similarly, PSA can be designed with fixed (non-adaptive) or dynamic (adaptive) link costs for the PSD. The diagram in Figure 2 shows the distribution of PSA problem types.

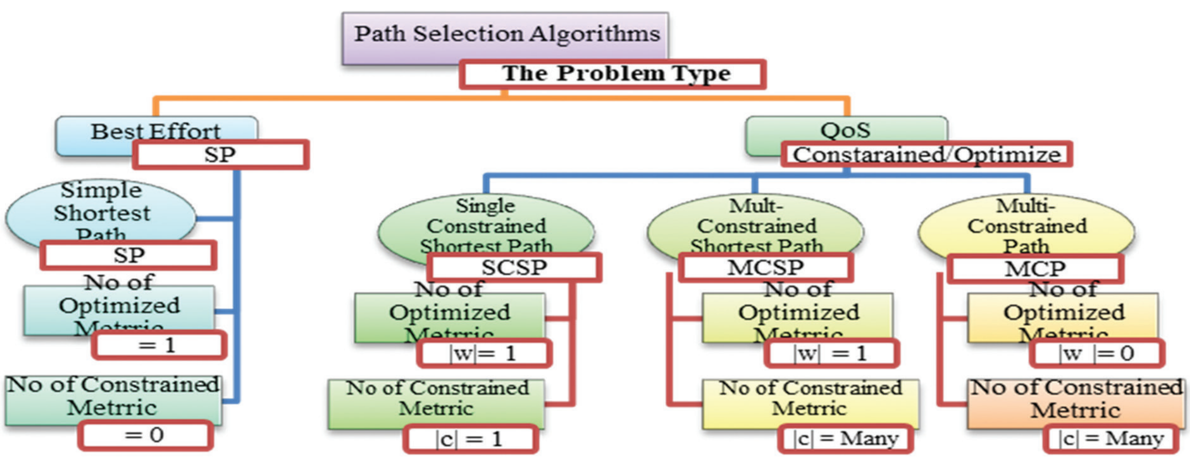

Fig. 2. PSA Problem Type

Best effort PSA does not aspire to guarantee that traffic will be delivered to its destination or meet any QoS requirements. The shortest path based on a minimum number of hop count with no other constrain is often used as the PS metrics. In contrast, PSA with QoS makes Path Selection Decisions (PSD) with predefined objectives. It aims to optimize some performance metric(s) while keeping others below a prescribed threshold. The metric to optimize is referred to as cost or weight. While the ones to be kept below a certain threshold are called constraints. Depending on requirements, the PSA can be designed to address different problems. For example, a PSA problem can be defined as a Single-Constrained Shortest Path (SCSP), a Multi-Constrained Shortest Path (MCSP), or a Multi-Constrained Path (MCP) Algorithm.

Consider a PSA for a network modelled as a graph $G=(S, E)$, with S, and E as a set of switches and communication links. The total number of switches and links are $|S|$ and $|E|$. If $w \in R_{+}^{|E|}$, is a vector that denotes the weight of the links between two adjacent switches $s \in S$. Let $c \in R_{+}^{M}$, denotes another vector to holds M elements corresponding to the threshold of the constrains metrics. Also, let $M \in R_{+}^{M x|E|}$, be a matrix that represents the values of the constraints for individual link $e_{ij} \in E$, between source i and destination j. Also, let P_{ij}, be the complete path between the source s_i and destination s_j, in the set of all paths P in G. With these factors, SCSP, MCSP, or MCP can be formulated mathematically to optimize any QoS metric (Min or Max) represented by the weight vector w as:

$$SelectP = min\, w^{T}X, \qquad \forall\, p_{ij} \in P \tag{1}$$

$$\mathcal{M}X \leq c \tag{2}$$

$$X = \begin{cases} 1, & if \quad e_{ij} \in p_{ij} \\ 0, & if \quad e_{ij} \notin p_{ij} \end{cases} \tag{3}$$

SCSP finds a route with minimum end-to-end QoS metric while keeping another metric below a prescribed bound. It corresponds to a situation where c in equation (2) = 0. In contrast, an MCSP is defined to optimize many end-to-end QoS metrics constrained by individual bounds. It corresponds to a situation where c in equation (2) >1. Lastly, an MCP is a selection problem defined without an optimization metric. The route to be selected while keeping some QoS metrics below a prescribed threshold. It corresponds to a situation where c in equation (2) > 1.

2.5. CLASSIFICATION OF PSA SOLUTIONS IN SDN

As shown in Figure 3, Path Selection Algorithms PSA considers link quality, switch role (critical node) flow characteristics, or a combination of these during path selection decision-making. This paper classifies the existing PSA based on these criteria.

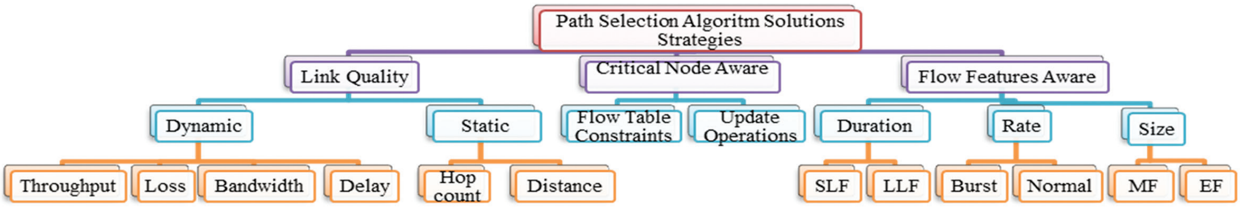

Fig. 3. Classification oF PSA in SDN

2.5.1. Link Quality

Link quality is estimated in teams of metrics such as delay, bandwidth, loss, throughput, or jitter. Depending on the problem, these metrics can be modelled as an objective function of an optimization problem while keeping other metrics as constraint(s). These metrics are either static or dynamic. Dynamic PSA uses metrics such as bandwidth. It is dynamic because the metric's value may change each time traffic is routed. In contrast, PSA, with a metric like a hop count, is static because it remains fixed unless there is a total change in topology [58]. The metrics are also classified as additive, multiplicative or non-additive. Additive metrics are delay, hops, and jitter. They imply that a path's end-to-end metric can be found accurately by summating the weight of different links. The weight of a path with non-additive metrics, like bandwidth, can only be established by the value of that constraint at the blockage link.

2.5.2. Critical Node(Switch)

A node/switch in a communication network is critical if many switches select it as the next hop because it falls along the paths of other switches. Consequently, because of its position, the communication frequency of such a switch with the CP for rule installation is higher than regular switches due to the number of SP passing through it. Such as, a switch tends to generate high communication overheads. However, the regular switches are responsive and dependent on the efficiency of the critical switch. Tools in graph theory such as degree [74], betweenness [75], information [76], closeness centralities, and PageRank [77] can be used to measure switch importance in a network.

Consider a network $G=(S, E)$, where S is a set of switches and E communication links. The network identifies a critical switch in terms of parameters such as switch update operations, switch flow table residual capacity or switch port's residual capacity[73]. The number of rules operations finds the switch update operation on each switch. The value reflects the total traffic load between different sources to destination switch pairs. For instance, a switch $s_{\Omega}\in S$ serving many intermediary switches through, $|P_{sp}|$ number of shortest paths between s_i to s_j switch pairs is considered more critical than regular switch $s_{\omega}\in S$. Switch s_{Ω} will be heavily loaded with a high number of rules in its flow table in comparison to s_{ω}. See Table (2) for a comparison of different measurement metrics.

Table 2. Comparison Table of Node Importance Evaluation Metrics

	Metric		Description
Node and edge Centrality	Eigenvectors **EV**		**EV** measures the importance of a node in a network. It is based on the principle that connections to nodes with a high degree contribute more to its score than connections to nodes having a low score.
	PageRank **PR**		The score computed by **PR** is higher for nodes that are highly connected with nodes that are highly connected themselves. **PR** score is iterated until convergence. **PR** is a variant of the EV centrality measure.
	Hyperlink-Induced Topic Search: **HITS**		• **HITS** calculates two scores: **Hub** and **Authority** • The more a node has outgoing links, the higher the Hub score. While the more a node has incoming links, the higher its **Authority** score. • Initially, every node is considered a **Hub** • **Authority** scores are fixed to a constant. The scores are updated and converge after a few iterations.
	Degree Centrality **DC**		A **DC**-based measure of individual centrality corresponds to how well-connected the individual is within their local environment.
	Closeness Centrality **CC**		**CC** measures centrality on a global scale based on how close a node is to all the other nodes. The idea is that a switch is central if it can interact with all others quickly
	Betweenness Centrality **BC**	Geodesic	**BC** of a node measures the extent of the role of the intermediary node in interacting with others.
		Path/flow	
		Random path	
	Information Centrality **IC**		**IC** metric is based on the concept of efficient propagation of information over the network. IC of a node is the relative drop in the network efficiency caused by the removal of the node from the network. It combines the idea of CC and BC measures.

2.5.3. Flow Characteristics

A flow has been defined differently based on the research context [3]. Most definitions convey that a flow is a set of packets sharing common identification properties while passing an observation point during a specific period [78]. These properties, in most cases, include five tuples like protocols such as (TCP, UDP, ICMP), source-destination IP, and port numbers.

Network traffic is made up of different types of flows [52]. Examples are EF and MF. EF are typically very few, about 1 -10 % of total network traffic. However, they are Long-Lived flows(LLF). As such, they tend to rapidly consume network buffers which consequently impose congestion and queuing delays to the minor majority of MF [4]. Each of these traffic types has specific QoS demand and behaves differently en route to destinations. Therefore, considering flow features during PSD is necessary because of their potential impact on network performance.

Flow statistics such as packet counts, bytes count, duration, per-flow packet size distribution (PSD), rate, or burst [81,82] are used to classify flows. These metrics can be used in isolation or combination to classify the flows using a threshold or threshold-less approach.

- Flow Size fS: is the number of bytes transmitted in a flow. It can be quantified using byte or packet count. With fS, flows are classified using a threshold Th such that if the $fS>Th$, then the flow is an EF and an MF otherwise.
- Flow Duration fD: This is the elapsed time between the first and the last packet of a flow. In a threshold-based, any flow with $fD<Th$ are tortoises, while flows with $fD\geq Th$ are dragonflies.
- Flow Arrival Rate fR is found by dividing the size fS of the transmitted data by the total flow duration $fD.fR=fS/fD$. If $fR>Th$, it is a cheetah and snail otherwise.
- Burst: Traffic burst investigates the extent of traffic and connection dominance in a network. Burst traffic involves packets with a short inter-arrival period. If packets inter-arrival time $iAT>Th$ are called porcupines and stingrays otherwise [78].

However, there is no unanimous settlement among the proposed flow classification approaches on what flow feature to adopt. However, the majority of the approaches used flow size [79], [84–87]. To set up the classification threshold, many other techniques used duration [88], rate [89–91], or burst as well. So, many PSA in SDN designed their solutions while considering these factors. These algorithms are reviewed in section 3.3 and summarised in Table 6

3. REVIEW OF PATH SELECTION ALGORITHMS IN SDN

3.1. PSA WITH LINK QUALITY AWARE

To choose a path for traffic in an SDN-Edge computing network, Hu et al.[92] considered loss and latency to design a Path Selection Method (PSM). The edge node at the SDN boundary is configured to assign network resources, such as bandwidth, according to flow requirements. Flows are directed via a path with lower packet loss. At the same time, delay-sensitive flows are routed via a path with minimal delay. Likewise, Alnajim and Salehi [93] proposed an incremental scheduling QoS-aware path selection technique to quickly redirect real-time applications with time-bound flows. The technique avoids bottleneck links causing scheduling impasses by selecting a path with enough residual bandwidth from the list of candidates' paths. Before the optimum PS, the technique incorporates an offline pre-routing where initial K paths are formed using Yen's [57]. The run time of this stage is high. The authors controlled it by coupling a Fibonacci Heap with Dijkstra [55] and [57]. However, in [92], [93], the switch load was not considered when choosing the path. Hence, QoS parameters like latency and the PDR should also be considered to ensure the quality of the link choice. In a similar technique in [31], switch port capacity in terms of data Transmission (Tx) and Receive Rate (Rx) is monitored. The data is pulled to determine the maximum capacity or load the port can accommodate. The statistics are fed to an application on a Floodlight [96]. The controller defines rules that send a flow via a port with Least Loaded Path LLP. In the validation, the authors used Mininet and Iperf to generate traffic. However, the constant gathering of statistics may significantly increase the controller overhead, thus increasing flow setup latency.

On the other hand, a QoS-driven and SDN-assisted Multipath Selection Scheme (QSMPS) was proposed in [94] to address the adaptability problem of traditional MPTCP. The method checks and assesses the network status using a scalable SDN- technique to gather statistics. Based on the data, an optimal number of sub-flows are found by QSMPS, which are distributed along the routes with short delays. The authors validate QSMPS in topology on Mininet using a Ryu framework. However, a best-case scenario based on residual bandwidth might not always ensure that distinct flows' requirements are met. Besides, the proposed scheme did not classify traffic according to its uniqueness. Because other flows can choose a short setup latency and a link quality that decreases the number of times links change due to topology changes. To compute the best paths while taking the QoS criteria for each flow into account, Saha et al.[95] introduced a greedy heuristic based on Yen's [57]. Multiple metrics are jointly considered in formulating the problem to get the best paths. The multi-constraint QoS-aware route is solved using Integer Linear Programming (ILP). The authors validate it on POX [96] using a Mininet with D-ITG [97] to generate IoT- traffic. However, in large networks with frequent topology changes, the ILP-based method may slow the convergence of the routing rules [88]. Perner and Carle [98] study the effects of various optimization on network link utilization and latency. A path selection technique was designed and bounded with some constraints to meet its requirements. One of the constraints is TCAM's size limitation. The constraint is modelled such that the number of outgoing flows does not exceed the maximum number of forwarding table entries. However, the switch update operation is not considered. Similarly, just like [87], the technique may suffer from high routing convergence time due to the ILP model.

Khalili et al. [99] designed a mechanism that determines a controller's flow setup latency by crafting and sending Special Ping Packets (SPP) between source-destination pairs. The Round-Trip Time RTT of the packets is measured. The time is attached to the individual path to get the total path setup latency. The technique is designed for scCP. Accordingly, Ravuri et al. [100] consider using dmCP to improve [99] overhead, scalability, and Single Point Failure SPOF experience in scCP. The paper emulates the DP on Mininet and implements the dmCP with floodlight[101], ONOS [102], and Kandoo [103]. However, both schemes may experience a high flow table operation, affecting path setup and switching time.

Intersection-based routing using SDN and Fog computing is proposed in [104] to address communication coverage holes in VANET. The controller's global network knowledge is leveraged to collect street score information from fog nodes and feed it into Dijkstra [54] to build the routing path. In a similar approach, [105] uses SDN to propose a dynamic routing to cope with the problem of frequent changes in a Flying Ad Hoc Network (FANET) of a drone (UAV) network. The technique Hybridizes a traditional OLSR with an SDN controller to perform topology discovery, statistics gathering, and route computation. OLSR and SDN alternate network control depending on the network status. However, [106] note that the conventional routing commonly used in DCN, like OSPF, incurs significant overhead with high convergence time. For this reason, controller-side Regular Topology Routing (cRetor) is proposed. cRetor differs from other topology–aware routing in compatibility with various other topologies. Including topology description language in the scheme ends the need for LLDP to run first. The action frees and relieves the scarce bandwidth and processing loads on the controller, respectively. The authors claim that the route calculation time of the technique is fast, which makes the convergence time shorter. The overheads and the failover performance are admirable. Alidadi et al. [109] proposed a low-complexity SDN-MPLS algorithm encompassing a quid pro quo between load balancing, hop count, and power consumption in mobile –SDN with restricted bandwidth during PS.

Table 3. Comparison of PSA with Link Quality Aware

Ref	Path Selection Criteria						Use Case				Design Objective		Solution Approach & BLA			Validation & Implementation Tools
	Parameters			No of Constraints												
	Link QoS	Critical Switch	Flow Features	MCP	SCSP	MC SP	Custom/ Others	DCN	WAN	IoT	LB	QoS	RU	AI	MM / Others	
[92]	✓	X	X	X	X	✓				✓		✓	✓	X	✓	Simulation
[93]	✓	X	X	X	✓	✓	Random	X	X	X	X	✓	X	X	Fibonacci, Yen,Dijkstra,	
[31]	✓	X	X	X	✓	X	Fat Tree	X	X	X	X	✓	X	X	Dijkstra[54]	Floodlight Mininet, Iperf
[94]	✓	X	X	X	✓	X	✓	X	X	X	X	✓	X	X	✓	Ryu,[53] NetworkX
[95]	✓	X	✓	✓	X	X	X	X	X	✓	X	✓	X	X	ILP, Yen	POX, Mininet [107]
[98]	✓	✓	X	X	X	✓	Critical	X	X	X	X	✓	✓	X	ILP	ITZ
[99]	✓	X	X	X	✓	X	csCP	X	X	X	X	✓	X	X	Polling method	Floodlight, Mininet [107]
[100]	✓	X	X	X	✓	X	dmCP	X	X	X	X	✓	X	X		Floodlight, ONOS, Kando
[104]	✓	X	X	X	✓	X	VANET	X	X	✓	X	✓	X	X	Dijkstra	SUMO,NS2[108]
[105]	✓	X	X	X	X	✓	FANET	X	X	✓	X	✓	X	X	Hybridized	
[106]	✓	X	X	X	X	✓	DCell	✓	X	X	X	✓	X	X	A*[58]	Floodlight, Mininet
[109]	✓	X	X	X	X	✓	Mobile	X	✓	X	✓	✓	✓	X	------	MIRA

3.1.1 Multipath PSA With Link Quality Aware

Megyesi et al. [110] proposed a mechanism for measuring Available Bandwidth (ABW) during path selection. The authors substitute the distance metrics of [54] with the link ABW to decide which path to select. The method is modelled as a max-flow problem to adapt to different use cases requiring the choice of the best available path among multipath. A Ford-Fulkerson algorithm is employed for the max-flow problem. Another technique is proposed by Dutra et al. [111]. Where, for each of the ingress traffic in DCN, the technique provides it with the required end-to-end bandwidth and effective use of the switches along the selected path. The action leads to a reduction in path use cost and execution time.

Similarly, in the work of Celenlioglu and Mantar [112], a pre-established multi-paths (PMP) between each source-destination switch is used to visualize an underlying DP topology to design a PSA with resource management through Admission Control (AC), Load Balancing(LB), and Path Resizing(PR). In another approach, [113] and [114] proposed GridFTP for the parallel transfer of a large amount of scientific data along multiple paths based on the Dijkstra[54]. Equally, using a multipath approach, Tariq and Bassiouni [115] extended their initial design of a QoS Aware Multipath (QAMO) [116] for a traditional optical network to SDN. The extension supports adaptive QoS differentiation with a priority factor for burst traffic and link state. It uses a Dijkstra [54] to find K paths between the source-destination pair.

Furthermore, [117] proposed a multipath forwarding approach for next-generation networks using SDN. The technique addresses the need for a conducive inter-networking environment for future data-centric applications. The technique exploits the edge diversity of transit ISPs available across many IXPs to propose cross-layer coordination with SDN flexibility. The authors use [118] to discover the initial K-shortest paths. Before invoking a route reconciliation and update strategy to re-evaluate the initial choice at the Control eXchange Authority (CXA) Controller implemented with Ryu[53] SDN framework. Likewise, in a similar effort, A Dynamic Multipath Scheduling Protocol (DMSP) for identifying and isolating congestion-susceptible links in DCN using SDN is demonstrated in [119]. DMSP split the flow traffic among multipath to reduce the congestion. However, the splitting is done identically among the available paths. To address the problem of the unequal split of DMSP [119]. Farrugia et al. [122] proposed a Globally Optimised Multipath Routing (GOMR) algorithm that splits a flow traffic equally among multiple paths using a stochastic mechanism. GOMR leverages the global knowledge of network topology and its traffic statistics to formulate the problem as linear programming LP to optimize per packet multipath routing proposed.

Table 4. Comparison Table for PSA with Multipath

Ref	Path Selection Criteria						Use Case				Design Objective			Solution Approach & BLA		Validation & Implementation Tools
	Parameters			No of Constraints												
	Link QoS	Critical Switch	Flow Features	MCP	SCSP	MCSP	Custom / Others	DCN	WAN	IoT	LB	QoS	RU	AI	MM / Other	
[110]	✓	X	X	X	✓	X	X	✓	X	X	X	✓	X	X	Dijkstra, Ford Fulkerson	Floodlight, Mininet, DITG
[111]	✓	X	X	✓	X	X	X	X	X	X	X	X	✓	X	X	
[112]	✓	X	X	✓	X	X	Intra Domain	X	X	X	✓	✓	✓	X	✓ MM	Floodlight, Mininet Ditg
[113] [114]	✓	X	X	X	✓	X	Virtual & Real Env	X	✓	X	X	✓	✓	X	Dijkstra	OvS
[115]	✓	X	X	X	✓	X	DCN	✓	X	✓	X	✓	X	X	Dijkstra	C++
[117]	✓	X	X	X	X	✓	Next-Gen ISP, IXPs	X	✓	X	X	✓	X	X	EPPStein KSP[118]	Ryu, Mininet
[119]	✓	X	X	X	X	✓	Fat Tree	✓	X	X	X	✓	✓	X	EPPStein KSP[118]	Ryu, Mininet
[120]	✓	X	X	✓	X	X	Butterfly, G´EANT	✓	✓	X	✓	✓	X	X	LP	Ns-3, GLPK, LEMON

3.2. PSA WITH CRITICAL SWITCH AWARE

Yan et al. [24] employed a TCAM-aware flow rerouting approach to address the fault tolerance problem. The authors formulate the problem as optimization with an objective function that finds a set of backup paths with TCAM and bandwidth as constraints. Solutions are sought using Forward Local Rerouting (FLR) and Backward Local Rerouting (BLR) heuristics. The two heuristics are used alternatively depending on the network state. However, the coexistence of the multiple heuristics at the controller may introduce an extra computational complexity. SwitchReduce[37] presents another intermediate switch state and controller participation technique. The technique is founded on wildcard identical action flows, RouteHeaders (first hop-based routing), and division of labour principles. SwitchReduce ensures the number of rules in the flow table is decided by the corresponding actions taken on flows going through it and does not increase linearly according to the flows. However, the technique is not adaptive to topology changes and has not been validated on large DCN.

In a similar effort, Perner and Carle [98] study the effects of various optimization objective functions on network performance metrics, such as link utilization and latency. Path selection technique bounded with some constraints is designed to meet the network requirements. One of the constraints considered is the TCAM size limitation. The constraint is modelled such that the number of outgoing flows does not exceed the maximum number of forwarding table entries. Astaneh et al. [121,122] proposed another work to restore SDN failures by rerouting disrupted flow with several switch update operations. The technique finds the switches with a small number of flow table entries to reroute the flows during the path restoration process. It is designed as a local restoration plan problem formulated as an ILP to trade-off between path cost and the switch update operation. Dijkstra [54] is leveraged to select a path with a minimum hop count. However, the technique may suffer from high routing convergence time due to the ILP. In another technique, Malik et al. [123], [124] proposed an alternative way to reduce switch update operation and preserve the limited size of the flow table during rerouting at the time of failure. They proposed an optimum path selection while looking at shared links across paths. The shared links have a higher tendency to bring down the consumption of flow table spaces. The technique picks a route with the most shared links from the paths list. it is validated with POX[96] on the Mininet. However, when the switch utilization rate rises, there is a greater likelihood of load imbalance and possible overflow.

Incidentally, Yu et al. [125] presented a path selection method based on node significance and flow prediction. The authors use Deep Neural Networks (DNNs) Q-Leaning [126] to balance the network load. The scheme comprises three algorithms running as the intelligent centre on the controller. The identification of the critical nodes is made using H-index. Flow forecast is achieved with DNN, and path selection is based on Q-leaning according to node importance. However, despite the benefits, the TCAM space constraint might limit the potential use of DNN[127] and Q-Leaning [126] that so much relies on historical data. In another work, Gotani et al. [128], [129] proposed a technique to reduce the effect of switch processing latency during path setup. Since the time to add flow entries is different for different switches. The paper designed a scheme of three methods to select an optimized path while minimizing the total switching time. The authors made path selection decisions based on path-switching delay in multiple paths with varying switch processing times. In this manner, path screening took place, and the one with the set of switches that requires the least processing time was selected. In a large-scale network, the solution might not deliver the best performance. Also, in the work of Isyaku et al. [1], [73] link quality and switch update operation is considered for path selection. However, the authors ignore the heterogeneity of the traffic traversing the network at the point of the path selection decision.

To minimize the impact of high demands of flow rule updates in space-hungry TCAM, the work in [130] considers the diversity of instruction types and switch behaviour to propose the ***RuleTailor*** algorithm. RuleTailor is an efficient, measurement-based optimization framework for SDN flow routing rules updates. Different from the consideration of switch behaviour in RuleTailor, the techniques proposed in [131,132] adopt the concept of aggregation of the routing rules. The approach aims to trim the number of possible rules in the table. [131] classify paths into two, one for popular flow and the other for non-popular flow. On the other hand, Jia and Wang [132] translate the Destination Address and Source-Port on Demand (DATSPToD) so that aggregate routing rules are taken to minimize entries in the flow table. The technique is meant for Large-scale SDN with scattered address space allocation. Similarly, DATSPToD is founded on address modification and port rewriting to address the problem of inefficient routing due to interleaved allocation of a non-contiguous IP address. Similarly, using Mixed Integer Non-Linear Programming (MINLP), Guo et al. [133] introduce path cardinality constraints on a PSA to prune the number of rules in TCAM. Dijkstra [54] is applied to find initial paths. Then passes them through a route optimization function before invoking an H-Permissible to prune and select a path based on the cardinality constraints. However, the inclusion of the optimization function adds up to the complexity by order of $(LP(N^2{+}L,\ L))/2$ [77]. Maaloul et al. [134] proposed a technique for SDN-based CGN to optimize energy. The controller dynamically turns on or off a network component while considering their residual space. The authors formulate the problem as Binary Integer Linear Programming (BILP) with TCAM and link usage as constraints. The aim is to minimize the power consumption of links and switches. A First-Fit Heuristics (FFH) is designed to solve the problem based on traffic demands. Dijkstra [54] is used as the baseline algorithm to identify, sort, and select paths according to First-Fit Most-Power (FFMP), First-Fit Least-Power (FFLP), and First-Fit Random (FFR).

Table 5. Comparison Table of PSA with Critical Switch Aware

Ref	Path Selection Criteria						Use Case				Design Objective			Solution Approach & BLA		Validation & Implementation Tools
	Parameters			No of Constraints												
	Link QoS	Critical Switch	Flow Features	MCP	SCSP	MCSP	Custom	DCN	WAN	IoT	LB	QoS	RU	AI	MM	
[24]	X	TCAM Size	X	✓	X	X	Failure									GLPK, Internet2
[36]	X	✓	X	X	X	X	ITZ	✓	X	X	X	X	✓	X		NoX OVS, Mininet
[98]	✓	TCAM Size	X	X	X	✓	Critical system	X	X	X	X	✓	✓	X	✓ ILP	ITZ
[121], [122]	X	Update Operation	X	X	X	✓	ITZ	X	X	X	X	X	✓	X	ILP, Dijkstra	ERnet USnet
[123], [124]	✓	Size & Update	X	X	✓	X	✓	X	X	X	X	✓	X	X	✓	POX Mininet NetworkX
[125]	X	Node Role	✓	X	X	✓	Random	X	X	X	✓	✓	X	DNN Q-L	X	---
[128], [129]	X	Switch CPU	X	✓	X	X	Disaster	X	X	X	X	✓	X	X	✓	NA
[1], [73]	✓	✓	X	X	X	✓	Custom	X	X	X	X	✓	✓	X	✓	Ryu, Mininet
[130]	X	Switch Role	✓	X	✓	X		X	X	X	✓	X	✓	X	✓	Ryu OVS, Iperf
[131]	X	TCAM compression	X	X	X	X	Flow rule update	X	X	X	X	X	✓	X	Aggregation Markov	Floodlight OVC MATLAB
[132]	X	TCAM	X	X	X	X		X	✓	X	X	X	✓	X	✓	NA
[134]	✓	TCAM power cost	X	X	✓	X	Energy CGN	X	X	X	✓	X	✓	X	BILP, Heuristic Dijkstra	SNDlib, CPLEX MATLAB
[133]	✓	TCAM	X	X	✓	X	ITZ	X	X	X	X	X	✓	X	MINLP Dijkstra,ARA	C++

3.3 PSA WITH FLOW CHARACTERISTICS AWARE

Several PSAs have been proposed to support the streaming of video flows [135–139]. Civanlar et al. [135] formulate the problem as Linear Programming (LP) to minimise weighted route length and packet loss. The technique finds the best path to accommodate video and the shortest path for the best effort. Harold et al. [136] designed the Routing Module (RM) of their proposed Video scheme based on the A* Prune [58]. The technique returns a list of paths that satisfy bandwidth, jitter, and delay constraints. It includes three modules for policy, admission control, and path reservation, with adequate resources for traffic control. However, the policies cannot guarantee the reservation of paths for all requests made. Thus, delay and packet loss might be experienced. The authors attribute the limitation to the scalability associated with a single controller. In a similar effort, [137] proposed an adaptive technique to reroute video with QoS support using LARAC [140]. The authors implement the algorithm on a Floodlight [101]. In a different approach, [138] improves the QoS provisioning of video services from the server side. The authors proposed a framework for load-balancing over a single-operator network to improve the QoS of video streaming. The framework monitors the load of the servers to track packet loss and delay variation. It redirects streaming requests using LARAC to a video server with a lighter load. In another work, a technique of PS for multi-media flows is proposed by Chooprateep et al.[139]. The authors designed a Video (VPSA) that finds a path based on Yen's algorithm using the controller's data of the link's past and present bandwidth utilization. Depending on whether a suitable path exists, video flows are either refused or allowed upon arrival. However, the coefficient used in the problem model is not adaptive to the characteristics of the traffics. Moreover, the algorithm considers only a single dynamic QoS metric while ignoring static metrics such as link latency and video holding.

In Egilmez et al. [20], an OpenFlow controller application, OpenQoS, implemented on a Floodlight [101], is proposed to handle multimedia traffic separately with end-to-end QoS assurance. OpenQoS is designed based on LARAC [140] approach. The algorithm uses the packet's header field in the MPLS structure of every incoming traffic to classify and separate multimedia traffic from data traffic. Network statistics are collected via feature_request messages every 1ses to enable the calculation of the QoS of each available route. With the separation and QoS knowledge of all routes, each traffic class is handled differently. The multimedia flows are placed on the route with the required QoS resources. At the same time, the data traffic is handled with the best effort forwarding. Saha [95] took the QoS metric specific to each flow into account to propose a greedy heuristic based on Yen's k-SP algorithm. The algorithm selects the ideal path for each flow. Multiple metrics are considered in formulating the problem as Integer Linear Programming (ILP). The model is validated using POX [96] controller and Mininet with D-ITG[97] to generate an IoT-based use case traffics. However, in large networks with frequent topology changes, using the ILP-based method may result in slow network convergence [88]. Kotani and Okabe [141], employed a packet filtering technique to separate the most critical flows from others at the level of packet-in messages. The filtering protects CP from a high packet rate and reduces the load on DP switches. An experiment reveals that with the mechanism, switches could significantly moderate the CPU loads, thereby preserving the space constraint of TCAM. However, the rate restriction mechanism in the technique presents some cases of packet lost and slight overhead. HiQoS [142] is also proposed by Jinyao et al. as a Multipath QoS solution. The PS scheme includes a module for the differential handling of flows with QoS requirements and a module that finds multiple paths based on a modified Dijkstra[54]. The controller uses the IP address of the source switch to separate several types of services and supplies diverse bandwidth assurances to each class. Bandwidth guarantee to specific traffic is achieved through the queuing mechanisms provided by the Openflow protocol [52]. HiQoS is bench marked with LiQoS and MiQoS.

The work in [143] proposed a framework for service differentiation support in SDN. It ensures the necessary QoS level for all multimedia applications. The approach leverages the controller's monitoring ability to get traffic statistics and network status every 3s. However, the controller might be overwhelmed with high overhead at this monitoring rate. The authors try to minimize that by restricting the statistics query to ingress switches only. In the work of Assefa and Ozkasap [144], a Machine Learning Framework for traffic aware energy efficient routing is proposed. The goal of MER-SDN is twofold, energy usage optimization and network performance. In a similar effort, Deng and Wang [145] applied Simulated Annealing (SA) Optimization to design a PSA to meet the specific QoS requirements of SDN-based IoT applications. AQRA classifies traffic into high, medium, and low priority using Class Identifier (QCI) obtained from the application profile set by the service providers. The initial path is determined using Dijkstra but updated later for each flow and placed in the switch flow table with rules designed by the SA-based routing module according to the application profile.

Table 6. Comparison of PSA with Flow Features Aware

Ref	Path Selection Criteria						Use Case / Aim				Design Objective			Solution Approach & BLA		Validation & Implementation Tools
	Parameters			No of Constraints												
	Link Quality	Critical Switch	Flow Features	MCP	CSP	MCSP	Custom	DCN	WAN	IoT	LB	QoS	RU	AI	MM	
[135]	X	X	✓	X	X	X	X	X	X	X	X	✓	X	X	LP	NOX
[136]	X	X	✓	X	✓	X	X	X	X	X	X	✓	X	X	A* Prune	FlowMonitorNS3
[137]	✓	X	✓	X	✓	X	✓	X	X	X	X	✓	X	X	LARAC	Floodlight Mininet
[138]	X	X	✓	X	X	X	X	X	X	X	✓	✓	X	X	LARAC	ODL
[20]	✓	X	✓	X	✓	X	X	X	X	X	X	✓	X	X	LARAC	Floodlight VLC MP
[95]	X	X	✓		X	✓	X	X	X	X	X	✓	X	X	Yen	X
[139]	✓	X	✓	X	✓	X	✓	X	X	X	X	✓	X	X	Yen	ITZ
[141]	X	✓	✓	X	X	X	X	X	X	X	✓	X	X	X	---	OVS
[142]	✓	X	✓	X	✓	X	✓	X	X	X	X	✓	X	X	Dijkstra	Floodlight, Mininet
[143]	✓	X	✓	X	✓	X	X	X	X	X	X	✓	✓	X	Dijkstra	POX
[144]	✓	X	✓	NA	NA	NA	X	NA	NA	NA	NA	NA	X	✓	NA	POX, SNDlib
[145]	✓	X	✓	X	X	✓	Campus	X	X	✓	✓	✓	X	SA	Dijkstra	Ryu, D-ITG

3.4. PSA WITH MACHINE LEARNING TECHNIQUES

Forecasting and classifying flow traffic is crucial to efficient resource utilization and QoS provisioning during PS for routing traffic in a modern network. The parameters included in the QoS specifications are usually captured in the SLA between service providers and subscribers. These parameters, described in section 2.4.2, are monitored and acquired from switches, ports, and flows using the OpenFlow built-in data collection module. Other metrics, such as residual bandwidth, link utilization, delay, and jitter, require extra effort to be acquired. There is a compelling need to efficiently measure these metrics and map each traffic with appropriate network resources to meet users' needs. Thus, additional intelligence is necessary to execute these tasks as desired. Thus, researchers have leveraged different Machine Learning (ML) techniques to synthesize network statistics controllers for traffic classification, routing, resource management, and load balancing. Traffic classification can be based on application or flow behaviours. The reason for using the former parameter is based on the need to separate Delay-Sensitive (DS) applications from Non-Delay-Sensitive (nDS). The DS application always required speedy detection and redistribution on the network to avoid SLA violations. However, with the wild upsurge of applications on the internet, it would be unrealistic to identify all the applications, especially in a large-scale network. The latter parameter help in separating EF from MF because the long-lived features of EF hurt the MF significantly. (See section 1.1 for detail). Different AI techniques can help detect, classify, and schedule each flows class as appropriate. Refer to Table 7 for a comparison summary of these techniques.

Cui and Xu [146] propose a PSA with load balancing in SDN based on multiple path features fed into Artificial Neural Network (ANN) model. ANN integrates the information and selects a path with a minimum aggregate load. The choice of the ANN to process the collected network statistics is due to its support for an infinite number of input vectors with undefined distribution in contrast to logistic regression and other probability methods. The technique goes through a Forward Propagation Learning (FPL) phase where the ABW, PLR, TL, and HC with a pre-set weight are supplied as input neurons. A Weight Adjustment (WA) phase is to adjust this weight until the fittest neural node is returned. This way, the controller finds a Least Loaded Path (LLP) to route traffic. Assefa and Ozkasap [144] propose a Machine Learning Framework for energy efficient routing and QoS optimization.

In a similar effort, Deng and Wang [145] applied Simulated Annealing (SA) to design a PSA to meet the specific QoS requirements of SDN-based IoT applications. AQRA incorporates a traffic classification module to categorize applications into high, medium, and low priority. The classification is according to QoS Class Identifier (QCI) obtained from the application profile set by the service providers. An initial path is determined using Dijkstra but updated later by SA for each flow. The SA-based module designs the routing rules according to the application profile. In another work [147], Energy Optimize Routing with Congestion Control for SDN WBAN is developed using Spider Monkey Optimization techniques. The network's weight/cost of available paths is modelled with residual energy level, link reliability, path loss, and queue length. Therefore, an optimum path among the paths is selected using the SMO algorithm. In a different approach, Naïve Bayes is used by El-Garoui et al. [148] to solve a routing problem. The solution optimizes Communication Overhead (CO) and Transmission Latency (TL) between pervasive nodes in SDN-VANET. The MLT influences CO reduction between the controller and RSU by predicting vehicle location as per RSU.

3.4.1. Genetic Algorithm (GA) Approach

Yu and Ke[149] acknowledge that video streaming is a pervasive killer application in the modern internet that require a highly efficient routing method to meet users' QoS demand. For this reason, they exploit the Genetic algorithm (GA) to develop a routing algorithm (GA-SDN) that can enhance video traffic over SDN. GA-SDN model the network as a connected graph with candidate paths from the source to the destination represented as (s_i, *list*, s_j). For any ingress traffic, the algorithm identifies a video in two ways; (1) ToS/DSCP bits of the packet and (2) Port number. If any packets whose information matches any video stream protocol, GA-SDN will not forward it according to the default SP. Instead, the algorithm will check the link utilization to determine whether the available bandwidth can provide the required QoS support. The technique is benchmarked against BF Algorithm[56]. Similarly,[150] deploys a secured GA-Based module in an SDN controller to perform a route calculation task that selects a path with optimized energy-consumed nodes in an IoT environment. Block-Chain technology is used to maintain a list of malicious activities of nodes in the DP, which the GA-Based routing module consider when taking a PS decision.

In contrast to GA-SDN, Li et al. [151] use Non-dominated Sorting Genetic Algorithm (NSGA II) to model a multi-objective optimization PS decision in SDN. It contains Monitoring (NMM), Awareness(NAM), and Reconfiguration (NRM) Modules. NRM receives instruction from NMM to reroute traffic to a better path when link utilization is high, or AVB is less than the flow requirements. The authors claimed that using NSGA II influences the reduction of forwarding latency and packet loss ECMP [6]. The work in [152] is another example of a routing problem solved with NSGA II. The authors applied the algorithm to propose a secure routing with untrusted DP switches.

3.4.2. Reinforcement Learning Approach

Reinforcement Learning (RL)[153] leverage the monitoring module of OpenFlow to gather comprehensive network statistic to build state representation space for the reward and action tuples to use in SDN. RL variants like Q-Leaning, Q-routing, and SARSA-Leaning have been applied in SDN to optimize different PS problems [154]–[159].

Conversely, [154] propose Q-FDBA to address video streaming problems related to Quality of experience (QoE) fairness. A Q-Leaning is used as the cluster decision algorithm to maximize the QoE. Other work by [155] combined State-Action-Reward-State-Action (SARSA)- with variable ε-Greedy function to solve SDN PS problems concerning congestion, packet queue waiting time, and transmission speed. In another approach, Huong et al. [157] combine RL with Deep Neural Network to develop a scheme called RLLP to address a load balance problem during PS. The reward function that guides the RLLP for load-balancing decisions considers the link delay, standard deviation, utilization, and discount rates. Whereas [158] uses RL components to propose a framework for traffic-aware energy-efficient routing HyMER. Similarly, in Shi et al. [159], the SARSA version of RL is also applied to develop a delay-aware PSA for SDN-supported power distribution application in an IoT environment (SDRS). SDRA uses an RL agent to adapt to the fluctuating network state and make a PS decision that improves system performance in terms of delay. Furthermore, an energy-efficient routing problem in large-scale SDN-IoT is also handled [160]. The authors propose two-level control mechanisms involving Multi-hop clustering MHC-RPL and a Q-Routing version of RL. However, in addition to its high convergence time, the requirement of a Q-learning algorithm to maintain a Q-table for storing *state*, *action*, and *reward* space information greatly limited its applicability to solving routing problems in SDN. Thus, to address the shortcomings of RL as experienced in Q_FDBA, Yu et al. [161] introduced a Deep Deterministic Policy Gradient (DDPG) mechanism to replace Q-table use with a neural network. The authors proposed Deep Reinforcement Learning (DROM) to optimize the routing procedure. DROM is benchmarked with OSPF concerning convergence time, delay, and throughput metrics.

Table 7. Comparison Table of PSA with Machine Learning

Ref	Path Selection Criteria						Use Case / Aim				Design Objective				Solution Approach & BLA		Validation & Implementation Tools
	Parameters			No of Constraints													
	Link Quality	Critical Switch	Flow Features	MCP	CSP	MCSP	IoT	DCN	WAN	Others	LB	QoS	FT	RU	AI	MM / Other	
[146]	✓	X	X	✓	X	X	X	✓	X	✓	✓	X	X	X	ANN		
[144]	✓	X	✓	X	X	X	X	✓	X	Energy	X	X	X	✓	✓	X	POX, Mininet SNDlib
[145]	✓	X	✓	X	X	✓	✓	X	X	Campus Network	✓	✓	X	X	SA	Dijkstra	Ryu, Mininet-WiFi, D-ITG
[147]	✓	X	X	✓	X	X	✓	X	X	WBAN	X	✓	X	✓	Spider Monkey		MATLAB
[148]	✓	X	X	X	✓	X	✓	X	X	VANET	X	✓	X	X	Naïve Bayes	X	Ryu, Mininet SUMO
[149]	✓	X	✓	✓	X	X	X	X	X	✓	X	✓	X	X	GA	X	Ns2, [108]myEvalSVC
[150]	X	✓	X	X	✓	X	✓	X	X	Security &Energy		✓	X	X	GA	X	MetaMask Ganache
[151]	✓	X	X	X	X	X	X	X	X	✓	✓	✓	X	✓	NSGA-II	X	Ryu, Iperf, Mininet
[152]	X	RL, Untruth	X	X	✓	X	X	X	X	Security	X	✓	X		NSGA-II	LP	Matlab
[154]	✓	X	✓	✓	X	X	X	X	X	✓	X	✓	X	X	Q-Leaning	X	ODL Mininet
[155]	✓	X	X	✓	X	X	X	X	X	✓	✓	✓	X	✓	SARSA & ε-Greedy	X	Mininet [107]
[157]	✓	X	X	✓	X	X	X	X	X	✓	✓	✓	X	✓	RL	X	Ryu, Mininet, DITG
[158]	✓	X	X	✓	X	X	X	✓	X	Energy	✓	X	X	✓	RL, Leaning	X	POX, MininetSNDlib
[159]	✓	X	X	X	✓	X	✓	X	X	PLC	X	✓	X	X	SARSA	X	--
[160]	✓	X	X	X	✓	X	✓	X	X	Energy	X	✓	X	✓	RL Q-Routing	X	Cooja, Contiki, RPL
[161]	✓	X	✓	X	✓	X	X	X	X	✓	X	✓	X	X	DRL	X	TensorFlow Keras,,OMNeT
[162], [163]	✓	✓	✓	X	✓	X	X	✓	X	X	X	✓	X	✓	DRL	X	OMNeT
[164]	✓	X	X	X	✓	X	X	✓	X	Backbone Network	X	✓	X	X	DRL, SA	X	OMNeT++
[165]	✓	X	X	X	✓	X	X	✓	X	Sprint	X	✓	X	X	DRL	X	OMNeT++
[166]	✓	X	X	X	X	✓	X	X	X	NSF NetARPANet	X	✓	X	X	DRL	Yen	Ryu, Mininet
[167]	✓	X	X	✓	X	X	✓	X	X	Security	X	✓	X	✓	DRL	X	Tensorflow
[168]	✓	✓	✓	X	X	✓	✓	X	X	FASNET		✓	X	X	ACO	X	TinyOS, MintRoute
[169]	✓	X	X	✓	X	X	X	✓	X	Energy	✓	✓	X	✓	ACO	LP	Floodlight, Mininet, Iperf3

The works in [162][163] also did a similar thing with DRL to address the problem of amalgamating EF with MF on the same path. The authors consider multiple network resources such as switch cache, link bandwidth to map, and schedule flow according to QoS requirements. In another effort, Maheswari et al. [164] also involved DRL to optimize routing procedures in SDN. The techniques aim to optimize network delay, network operation, and maintenance costs. The Traffic Matrix (TM), link weight, and network delay are represented by the state, action, and reward tuples. Likewise, in the work of Xu et al. [165], a DRL technique is integrated into SDN PSA to optimize performance concerning delay, hop count, and throughput. Chen et al. [166] formulated a traffic engineering problem in SDN. They developed an RL-Routing, based on DRL to find an optimized solution. Network delay and throughput are modelled in the state representation space of DRL. The reward function taps these metrics from this space to build an action space. The action space comprises a list of all paths and their associated cost (*reward*). Furthermore, the vulnerability of PSA in SDN to dynamic change of flow control rules at the time of malicious activities motivates Gou [167] to propose a DRL-based QoS- Aware security routing algorithm (DQSP) for IoT applications. DQSP is modelled to be immune to Gray Hole Attacks (GHA) and DDoS attacks. DQSP is evaluated in terms of PDR, E2E Delay, and probability of path attack (PA).

However, the *state* space of RL is overpopulated with many metrics, whose extraction and calculation from the MM might overwhelm the controller.

3.4.3. Ant Colony Optimization Approach

Ant-Colony Optimization (ACO) technique is exploited to develop a Traffic Differentiated Routing (TDR) [168]. The authors formulate a transmission reliability and prediction model as an LP. The model considers link availability and node forwarding ability to develop a TDR to guarantee the QoS of Flying Ad-hoc Sensor Networks (FASNETs). The also model seeks to arrive at a spanning tree that minimizes the average delay and improves data integrity for reliability-sensitive applications. The NP-hard nature of the problem compels the authors to seek a solution from the ACO. Torkzadeh et al. [169] incorporate an energy optimization constraint for a load-balancing routing problem in SDN. The authors propose a two-phase solution to solve the problem. A minimum graph ACO is employed in the first phase to prune the network topology. All inconsequential DP switches are discharged during routing, leaving only an energy-minimized sub-graph. In the second phase, a QoS-weighted PS technique is developed to route the traffic along paths with a balanced load based on a dynamic threshold value.

4. OPEN CHALLENGES FOR FUTURE RESEARCH

4.1. PSA WITH LINK QUALITY AWARE

4.1.1. Network State Information Problem

All PSA considering link quality parameters, whether static or dynamic (See section 2.4 for details), depend on Network State Information (NSI) for the PSD. The controller collects the NSI from the DP at time intervals and feeds it into the PSA to make the decision. PSA assumes this information to be accurate and adequate to make the right decision. The NSI is collected through sampling or polling techniques. However, if the former technique is used, the NSI might be inadequate or inaccurate at the time of PSD. However, it is a big challenge to maintain a high level of statistic collection accuracy in practice due to the periodic manner of the collection. Depending on the collection interval, the PSA might be called upon to take PSD during this interval. In this circumstance, the PSA must use the old information available. However, this might mislead PSA to make false positive or false negative decisions. Selecting the sampling period depends on the topology size and density if an overhead reduction is critical in the network. Therefore, it will be interesting to undertake further study to explore how the NSI collection period by the controller can dynamically adapt to topology and traffic changes. The study should provide a balance between adequate NSI for accurate PSD and message collection overhead on the controller. The idea of modelling link cost as a probabilistic metric is an exciting possibility in addressing inaccurate NSI.

Furthermore, it is essential to note that PSA with multiple constraints might not necessarily be an NP-complete problem. Instead, it further depends on other factors, such as topology size. Therefore, solution searching using an exact algorithm approach should be able to distinguish the scenario for which the complexity of the problem is polynomial to refine the solution searching approach.

4.1.2. Service Level Agreement (SLA) Evaluation Inconsistency

Subjectivity in evaluating satisfaction and compliance level of the QoS requirement captured in SLA by customers and service providers affects the trust and future relationship between them. The situation might sometimes lead to substantial financial loss for both parties [5]. This scenario is possible due to the lack of a QoS satisfaction measurement framework that can meticulously verify, audit, and validate the guarantee level pledged to each as defined in the SLA. Therefore, it is an exciting research concept to imagine a quantitative and unified technique to objectively provide a detailed evaluation of such agreement. Incorporating AI techniques might be helpful.

4.2 PSA WITH CRITICAL SWITCH AWARE

Several initiatives for different use cases have been proposed in recent years to solve a path selection problem for SDN. However, only a few have examined how SDN performs in dynamic, large-scale telecommunications networks, where heavy traffic flows are constantly generated. Similarly, the limited switch memory influences SDN performance in large-scale networks due to increased update operations and security-related risks. It is proven that in such an environment, many flows carry many packets in a short amount of time. Switch memory cannot hold the necessary amount of flow entries. This limitation remains one of the research problems that need additional studies. i.e., managing massive flows with many packets in the power-hungry tiny switch flow table.

4.2.1. Rule Update Operation on Switch flow table

Flow-table TCAM uses exact and wildcard matching rules for update operations [1]. Therefore, an effective PSA should understand the suitability of each possibility while formulating an optimization objective. Unlike exact rules, which must provide individual flow rules for each entry in a switch flow table. On the other hand, the wildcard rules allow multiple flows to be composed as one. This way, all identical entries can be recycled among various flows, thereby minimizing the number of entries and the overhead of frequent flow setup requests. PSA should also consider that TCAM operates slowly during an update operation. Therefore, packets may experience delays, especially in large networks. Hence, failure recovery must strictly comply with the CGN latency requirements.

4.2.2. Reactive and Proactive Rule Installation Hybridization

The number of critical switches in a network can be reduced by hybridizing reactive and proactive flow rule installation approaches. Employing a proactive approach for time-constraint applications is a preference. In contrast, best-effort or applications without deadline violation can embrace the reactive approach. For PSA to effectively utilize the limited TCAM space and decrease overhead and packet delay, efficient flow rule allocation should incorporate both reactive and proactive approaches. Therefore, designing a PSA while taking traffic variations along with these approaches will be an intriguing research topic. Re-routing rules should be executed in under 25ms to satisfy the strict QoS specifications of real-time applications.

4.3. PSA WITH FLOW CHARACTERISTICS AWARE

Traffic management has become essential to computer network design requirements [170]. The concept impacts various computer network areas of concern. E.g., Security issues such as intrusion and violation detection, glitch discovery such as TCP incast [171], and anomaly tracking such as DDoS [172]. Other areas are Traffic Engineering (TE), Quality of Service, Resource Management, Energy, PS Optimization and flow rerouting, Service Level Agreement (SLA) Appraisal, and Auditing. Most of the PSA with traffic awareness involves detecting and classifying a flow according to priority, size, or duration. The following issues are areas of concern that call for further investigation.

4.3.1. Flow Feature Selection Dilemma

Flow statistics such as size, duration, rate, or burst are used to compare a flow against a pre-defined threshold to classify flows accordingly. The threshold value can be fixed or adaptive. Selecting the flow feature for flow classification depends on the design objectives of the problem. There is no unanimous settlement on

4.3.2. Threshold Value Determination Challenge

The selection of a threshold value by most of the existing flow-aware PSA lacks a specific and systematic justification. Both static and adaptive threshold values run into this problem. Most of the papers reviewed in section 3.3 cited previously published works, for which, at the end of the citation chain, there is no justification for the chosen threshold. That trend is observed across numerous works. As a result, rather than being systematic, threshold selection in existing studies appears ad hoc. A study in [79] suggests that the preconfigured fixed threshold parameter for EF detection incurred high detection error rates because of its ability to adapt dynamically in real-time to constant traffic variability in a contemporary network. The tendency to report false positive and false negative errors is significant. The techniques do not adequately consider the dynamics of network traffic. Instead of being static, network traffic is dynamic. It may alter over time in response to variables such as the time of day, configuration changes, failures, or adjustments to the topology and instrumentation. Every time the threshold needs to be adjusted, the classification must be repeated to reflect the dynamic nature of the network. Manual threshold adjustment is nonetheless impossible due to the non-deterministic and frequent changes in network traffic conditions.

4.3.3. Flow Identification Overhead and Accuracy challenge

Flow detection mechanisms incorporated in PS schemes incurred some overhead in the stage of statistics gathering. Statistics-gathering techniques can be through sampling, polling, triggering, or hash functions. Each of these techniques has specific strengths and weaknesses regarding application areas and scenarios.

Sampling is one of the most adopted methods to acquire and profile networks for traffic analysis. Popular sampling method such as sFlow [35] has been incorporated by several (EFDM). It is scalable and adaptable to traffic heterogeneity in DCN [79], as several switches can be monitored efficiently by the sFlow protocol. In contrast, polling techniques deal with every flow entry in the flow table. A prior study [79] reveals that 64kb or 88bytes messages will return to the controller for each polling request. This data might not seem much, but if the average DCN size with 100 edge switches is factored in, that can reach up to 10mf/s on average. Summing up that for 88bytes, the total data to return will be up to 65GB. This data will dominate the limited bandwidth of the northbound interface. Therefore, employing this method to gather network statistics for EF identification in DCN will cause significant overhead and bandwidth mismanagement. Ref [4] suggests that EF are few in a DCN; thus, it is inefficient to accumulate data on each flow to detect EF in the network.

On the contrary, the triggering approach sets up a sniffer agent or applications at the end host [84]. The technique detects and classifies flows before transmission directly and precisely. Once the dimensions of a flow (e.g., socket buffer, flow size) surpass a set-up threshold, the EFDM decides that the flow is indeed an EF. The method reduces the overhead. However, it is impractical in DCN due to the requirement of changing the operation of each end host. For these reasons, it is interesting to conduct further studies in that direction.

4.4. PSA WITH MACHINE LEARNING TECHNIQUES

One of the significant challenges of employing ML techniques to solve real-life problems is the data set availability for model training. Privacy and confidentiality issues associated with computer networks make sharing this data difficult and scarce. The situation is worse in SDN because the technology is still emerging. For this reason, future research should be directed toward building and expanding the existing Opensource data set, such as SDN ITZ.

Secondly, PSA sought through ML should be adaptable to factors like communication mediums and applicable to technologies like Low-power and lossy network (LLNs) use in WSN. The forms of communication such as unicast, multicast, or broadcast applications (use case) like fog computing, DCN, and 5G. Likewise, traffic heterogeneity should be considered along with prediction patterns to guide the adaptation. Thus, NSI must be obtained regularly for accurate prediction and PS policy formulation. Thirdly, the ML model should incorporate safe mechanisms to preserve NSI integrity. This mechanism inclusion is necessary to avoid inaccurate findings or inconsistent decisions because the data collection could be impeded or even altered.

4.4.1. Training Dataset Scarcity Problem

One of the significant challenges of employing ML to solve real-life problems is the data set availability for model training. Privacy and confidentiality issues associated with computer networks make sharing this data difficult and scarce. The situation is worse in SDN because the technology is still emerging. For this reason, future research should be directed toward building and expanding the existing Opensource data set, such as SDN ITZ.

4.4.2. Intelligent Flow Table Management

In large networks, many flows arrive regularly, necessitating the installation of relevant rules in a flow table to occupy substantial storage space. Most SDN TCAM-related solutions in the literature are only evaluated on small networks [173]. What works for these networks cannot be compared to the number of devices in an extensive network such as WAN, DCN, or (IoT). ML techniques are handy for managing devices like switches and controllers. Many ML approaches, however, concentrate on flow classification and flow monitoring. Most research focuses on selecting the optimum traffic flow to be installed in advance rather than forecasting traffic flow for real-time applications and best-effort traffic. OpenFlow provides built-in data collection that stores flow-statistic like packet counts. This statistical data shows how frequent traffic flows. It will be interesting to develop a plan that takes advantage of this built-in data collection to reduce TCAM space usage. So the flow matching rate can speed up. Perhaps by applying fuzzy theory in the choice of recurrently used flow rules to be placed in the flow table.

5. CONCLUSION

SDN provides flexibility in managing the complexity and demand of our modern network, which was unable to be provided by traditional architecture. In this paper, we picked the network management task of path selection for routing traffic and reviewed the existing algorithms under four categories. (1) To guide their PSD, the PSAs with static link quality under different traffic conditions or dynamic link quality. (2) The PSAs that evaluate the criticality of a switch in terms of an update operation, flow table, and port capacity to guide PSD. (3) The PSAs that consider the traffic flow heterogeneities in terms of size, duration, burst, and priority to guide PSD. (4) The PSAs that use ML for PSD decisions. For each category, the papers were reviewed considering their path selection criteria, use case, design objectives, solution approach, baseline algorithms, and validation approaches. A comparison summary table is given at the end of each category. Based on the review, some persistent challenges related to each category are identified and recommended for further study. For instance, inaccurate and inadequate NSI in PSA with dynamic link quality and SLA evaluation are challenges that need further study. Secondly, rule update operations overhead is a challenge in PSA with critical switch awareness. The paper suggests hybridizing reactive and proactive rule installation approaches to optimize PSA convergence. Thirdly, the paper identifies flow feature selection and threshold value challenges during flow classification. Furthermore, the paper identifies flow identification overhead and accuracy as issues requiring further research efforts associated with PSAs considering traffic dynamics. Lastly, the paper identifies training dataset scarcity as a problem faced by PSA employing ML.

6. REFERENCES

[1] B. Isyaku, K. A. Bakar, M. S. M. Zahid, E. H. Alkhammash, F. Saeed, F. A. Ghaleb, "Route path selection optimization scheme based link quality estimation and critical switch awareness for software-defined networks", Applied Sciences, Vol. 11, No. 19, 2021,

[2] Z. R. Alashhab, M. Anbar, M. M. Singh, Y. B. Leau, Z. A. Al-Sai, S. A. Alhayja'a, "Impact of Coronavirus Pandemic Crisis on Technologies and Cloud Computing Applications", Journal of Electronic Science and Technology, Vol. 19, No. 1, 2021, pp. 25-40.

[3] B. Isyaku, M. S. Mohd Zahid, M. Bte Kamat, K. Abu Bakar, F. A. Ghaleb, "Software Defined Networking Flow Table Management of OpenFlow Switches Performance and Security Challenges: A Survey", Futur. Internet, Vol. 12, No. 9, 2020, p. 147.

[4] C. Y. Lin, C. Chen, J. W. Chang, Y. H. Chu, "Elephant flow detection in datacenters using OpenFlow-based Hierarchical Statistics Pulling", Proceedings of the IEEE Global Communications Conference, Austin, TX, USA, 8-12 December 2014, pp. 2264-2269.

[5] S. Khan, F. K. Hussain, O. K. Hussain, "Guaranteeing end-to-end QoS provisioning in SOA based SDN architecture: A survey and Open Issues", Future Generation Computer Systems, Vol. 119, 2021, pp. 176-187,

[6] C. Hopps, "Analysis of an Equal-Cost Multi-Path Algorithm", Network Working Group C. Hopps Request for Comments: 2992 NextHop Technologies Category: Informational, 2000, pp. 130-139.

[7] M. Al-Fares, A. Loukissas, A. Vahdat, "A scalable, commodity data centre network architecture", Proceedings of the ACM SIGCOMM Conference on Data Communication, 2008, p. 63

[8] H. Balakrishnan, M. Stemm, S. Seshan, R. H. Katz, "Analyzing stability in wide-area network performance", Performance Evaluation Review, Vol. 25, No. 1, 1997, pp. 2-12.

[9] M. Casado et al. "Rethinking enterprise network control", IEEE/ACM Transactions on Networking, Vol. 17, No. 4, 2009, pp. 1270-1283.

[10] A. Gupta et al. "An industrial-scale software defined internet exchange point", Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation, 2016, pp. 1-14 .

[11] I. Z. Bholebawa, R. K. Jha, U. D. Dalal, "Performance Analysis of Proposed Network Architecture: OpenFlow vs Traditional Network", Wireless Personal Communications, Vol. 86, No. 2, 2016, pp. 943-958,

[12] J. M. Jaffe, F. H. Moss, "A Responsive Distributed Routing Algorithm for Computer Networks", IEEE Transactions on Communications, Vol. 30, No. 7, 1982, pp. 1758-1762.

[13] R. Gallager, "A Minimum Delay Routing Algorithm Using Distributed Computation", IEEE Transactions on Communications, Vol. 25, No. 1, 1977, pp. 73-85.

[14] J. M. Mcquillan, I. Richer, E. C. Rosen, "The New Routing Algorithm for the ARPANET", IEEE Transactions on Communications, Vol. 28, No. 5, 1980, pp. 711-719.

[15] A. A. Neghabi, N. J. Navimipour, M. Hosseinzadeh, A. Rezaee, "Load Balancing Mechanisms in the Software Defined Networks: A Systematic and Comprehensive Review of the Literature", IEEE Access, Vol. 6, 2018, pp. 14159-14178.

[16] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, S. Uhlig, "Software-defined networking: A comprehensive survey", Proceedings of IEEE, Vol. 103, No. 1, 2015, pp. 14-76.

[17] M. Ejaz Ahmed, H. Kim, "DDoS attack mitigation in the internet of things using software-defined networking", Proceedingd of the IEEE Third International Conference on Big Data Computing Service and Applications, Redwood City, CA, USA, 2017, pp. 271-276.

[18] Q. Gao, W. Tong, S. Kausar, L. Huang, C. Shen, S. Zheng, "Congestion-aware multicast plug-in for an SDN network operating system", Computer Networks, Vol. 125, 2017, pp. 53-63.

[19] P. Yi, T. Hu, Y. Hu, J. Lan, Z. Zhang, Z. Li, "SQHCP: Secure-aware and QoS-guaranteed heterogeneous

controller placement for software-defined networking", Computer Networks, Vol. 185, No. 2020, p. 107740.

[20] H. E. Egilmez, S. T. Dane, K. T. Bagci, A. M. Tekalp, "OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks", Proceedings of the Asia Pacific Signal and Information Processing Association Annual Summit and Conference, Hollywood, CA, USA, 3-6 December 2012.

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

[22] M. W. Hussain, B. Pradhan, X. Z. Gao, K. H. K. Reddy, D. S. Roy, "Clonal selection algorithm for energy minimization in software-defined networks", Applied Soft Computing, Vol. 96, 2020, p. 106617.

[23] Z. N. Abdullah, I. Ahmad, I. Hussain, "Segment routing in software-defined networks: A survey", IEEE Communications Surveys & Tutorials, Vol. 21, No. 1, 2019, pp. 464-486.

[24] P. Murali Mohan, T. Truong-Huu, M. Gurusamy, "Fault tolerance in TCAM-limited software-defined networks", Computer Networks, Vol. 116, 2017, pp. 47-62.

[25] B. Yan, Q. Liu, J. Shen, D. Liang, B. Zhao, L. Ouyang, "A survey of low-latency transmission strategies in software-defined", Computer Science Review, Vol. 40, 2021, p. 100386.

[26] J. Mei, K. Zheng, L. Zhao, Y. Teng, X. Wang, "A Latency and Reliability Guaranteed Resource Allocation Scheme for LTE V2V Communication Systems", IEEE Transactions on Wireless Communications, Vol. 17, No. 6, 2018, pp. 3850-3860.

[27] V. Seedha Devi, T. Ravi, S. B. Priya, "Cluster Based Data Aggregation Scheme for Latency and Packet Loss Reduction in WSN", Computer Communications, Vol. 149, No. 2019, pp. 36-43.

[28] Z. Cai, A. Cox, E. T. S. Ng, "Maestro: A System for Scalable OpenFlow Control", Cs.Rice.Edu, 2011 p. 10.

[29] N. Gude, J. Pettit, B. Pfaff, N. Mckeown, S. Shenker, "NOX : Towards an Operating System for Networks", ACM SIGCOMM Computer Communication Review, Vol. 38, No. 3, 2008, pp. 105-110.

[30] L. Zhu, M. M. Karim, K. Sharif, F. Li, X. Du, M. Guizani, "SDN Controllers: Benchmarking & Performance Evaluation", arXiv:1902.04491, 2019.

[31] M. F. Rangkuty, R. Muslim, T. Ahmad, M. H. A. Al-Hooti, "Path selection in software-defined network data plane using a least loaded path", Proceedings of the International Conference on Advanced Computer Science and Information Systems, Depok, Indonesia, 17-18 October 2020, pp. 135-140.

[32] E. T. B. Hong, C. Y. Wey, "An optimized flow management mechanism in OpenFlow network", Proceedings of the International Conference on Information Networking, Da Nang, Vietnam, 11-13 January 2017, pp. 143-147.

[33] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, S. Banerjee, "DevoFlow: Scaling flow management for high-performance networks", Computer Communications Review, Vol. 41, No. 4, 2011, pp. 254-265.

[34] M. Yu, J. Rexford, M. J. Freedman, J. Wang, "Scalable flow-based networking with DIFANE", ACM SIGCOMM Computer Communications Review, Vol. 40, No. 4, 2010, pp. 351-362.

[35] P. Phaal, "Traffic Monitoring using sFlow", www.sflow.org/sFlowOverview.pdf. (accessed: 2022)

[36] A. S. Iyer, V. Mann, N. R. Samineni, "SwitchReduce: Reducing switch state and controller involvement in OpenFlow networks", Proceedings of the IFIP Networking Conference, Brooklyn, NY, USA, 22-24 May 2013, pp. 1-9.

[37] E. Akin, T. Korkmaz, "Comparison of Routing Algorithms with Static and Dynamic Link Cost in Software Defined Networking (SDN)", IEEE Access, Vol. 7, 2019, pp. 148629-148644.

[38] F. Y. Okay, S. Ozdemir, "Routing in Fog-Enabled IoT Platforms: A Survey and an SDN-Based Solution", IEEE Internet of Things Journal, Vol. 5, No. 6, 2018, pp. 4871-4889.

[39] M. Karakus, A. Durresi, "Quality of Service (QoS) in Software Defined Networking (SDN): A survey", Journal of Network and Computer Applications, Vol. 80, no, 2016, pp. 200-218.

[40] S. Tomovic, I. Radusinovic, N. Prasad, "Performance comparison of QoS routing algorithms applicable to large-scale SDN networks", Proceedings of the International Conference on Computer as a Tool, Salamanca, Spain, 8-11 September 2015.

[41] J. W. Guck, A. Van Bemten, M. Reisslein, W. Kellerer, "Unicast QoS Routing Algorithms for SDN: A Comprehensive Survey and Performance Evaluation", IEEE Communications Surveys & Tutorials, Vol. 20, No. 1, 2018, pp. 388-418.

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

[43] M. Hamdan et al. "A comprehensive survey of load balancing techniques in a software-defined network", Journal of Network and Computer Applications, Vol. 174, No. 2020, 2020, p. 102856.

[44] F. Chahlaoui, H. Dahmouni, "A Taxonomy of Load Balancing Mechanisms in Centralized and Distributed SDN Architectures", SN Computer Science, Vol. 1, No. 5, 2020, pp. 1-16.

[45] B. Heller, R. Sherwood, N. Mckeown, "The controller placement problem", Computer Communications Review, Vol. 42, No. 4, 2012, pp. 473-478.

[46] J. Cui, Q. Lu, H. Zhong, M. Tian, L. Liu, "A Load-Balancing Mechanism for Distributed SDN Control Plane Using Response Time", IEEE Transactions on Network and Service Management, Vol. 15, No. 4, 2018, pp. 1197-1206.

[47] R. Wazirali, R. Ahmad, S. Alhiyari, "Sdn-OpenFlow topology discovery: An overview of performance issues", Applied Sciences, Vol. 11, No. 15, 2021.

[48] R. Amin, E. Rojas, A. Aqdus, S. Ramzan, D. Casillas-Perez, J. M. Arco, "A Survey on Machine Learning Techniques for Routing Optimization in SDN", IEEE Access, Vol. 9, 2021, pp. 104582-104611.

[49] M. F. Tuysuz, Z. K. Ankarali, D. Gözüpek, "A survey on energy efficiency in software-defined networks", Computer Networks, Vol. 113, 2017, pp. 188-204.

[50] B. G. Assefa and Ö. Özkasap, "A survey of energy efficiency in SDN: Software-based methods and optimization models", Journal of Network and Computer Applications, Vol. 137, No. 2018, 2019, pp. 127-143.

[51] F. Dabaghi, Z. Movahedi, R. Langar, "A survey on green routing protocols using sleep-scheduling in wired networks", Journal of Network and Computer Applications, Vol. 77, No. 2016, 2017, pp. 106-122.

[52] N. McKeown et al., "OpenFlow", ACM SIGCOMM Computer Communications Review, Vol. 38, No. 2, 2008, pp. 69-74.

[53] Course Hero, "Lab 4-1-Ryu Introduction" (accessed: 2022)

[54] E. W. Dijkstra, " A note on two problems in connexion with graphs", Numerische Mathematik, Vol. 271, 1959, pp. 269-271.

[55] P. E. Hart, N. J. Nilsson, B. Raphael, "Formal Basis for the Heuristic Determination eijj ", Systems Science and Cybernetics, Vol. 4, No. 2, 1968, pp. 100-107.

[56] R. Bellman, "On a Rooting Problem", 1956.

[57] J. Y. Yen, " Finding the K Shortest Loopless Paths in a Network", Management Science, Vol. 17, No. 11, 1971, pp. 712-716.

[58] G. Liu and K. G. Ramakrishnan, "A *Prune: An algorithm for finding K shortest paths subject to multiple constraints", Proceedings of the Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society, Anchorage, AK, USA 22-26 April 2001, pp. 743-749

[59] R. K. Ahuja, "Network Flows", Indian Institute of Technology, 1993.

[60] M. R. Garey, D. S. Johnson, "Computers, Complexity, and Intractability", Computers and Intractability: A Guide to the Theory of NP-Completeness, 1979, p. 115.

[61] M. I. Hamed, B. M. ElHalawany, M. M. Fouda, A. S. T. Eldien, "A novel approach for resource utilization and management in SDN", Proceedings of the 13th International Computer Engineering Conference, Cairo, Egypt, 27-28 December 2018, pp. 337-342, ,

[62] N. Thazin, K. M. Nwe, Y. Ishibashi, "Resource Allocation Scheme for SDN-Based Cloud Data Center Network", Procedings of ICCA, 2019 pp. 15-22.

[63] M. Jeon, N. Kim, Y. Jang, B. D. Lee, "An efficient network resource management in SDN for cloud services", Symmetry., Vol. 12, No. 9, 2020.

[64] D. Erickson, “The Beacon OpenFlow controller”, Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, August 2013, pp. 13-18.

[65] M. Khorramizadeh and V. Ahmadi, “Capacity and load-aware software-defined network controller placement in heterogeneous environments”, Computer Communications, Vol. 129, 2018, pp. 226-247.

[66] M. T. I. Ul Huque, W. Si, G. Jourjon, V. Gramoli, “Large-Scale Dynamic Controller Placement”, IEEE Transactions on Network and Service Management, Vol. 14, No. 1, 2017, pp. 63-76.

[67] W. H. F. Aly, “Generic Controller Adaptive Load Balancing (GCALB) for SDN Networks”, Journal of Computer Networks and Communications, Vol. 2019, 2019, pp. 1-9.

[68] K. S. Sahoo et al., “ESMLB: Efficient Switch Migration-Based Load Balancing for Multicontroller SDN in IoT”, IEEE Internet of Things Journal, Vol. 7, No. 7, 2020, pp. 5852-5860.

[69] W. H. F. Aly, “Generic Controller Adaptive Load Balancing (GCALB) for SDN Networks”, Journal of Computer Networks and Communications, Vol. 2019, 2019, pp. 1-9.

[70] A. Mahjoubi, O. Zeynalpour, B. Eslami, N. Yazdani, “LBFT: Load Balancing and Fault Tolerance in distributed controllers”, Proceedings of the International Symposium on Networks, Computers and Communications, June 2019, pp. 1-6.

[71] H. Chen, G. Cheng, Z. Wang, “A game-theoretic approach to elastic control in software-defined networking”, China Communications, Vol. 13, No. 5, 2016, pp. 103-109.

[72] C. Wang, B. Hu, S. Chen, D. Li, B. Liu, “A Switch Migration-Based Decision-Making Scheme for Balancing Load in SDN”, IEEE Access, Vol. 5, 2017, pp. 4537-4544.

[73] B. Isyaku, K. bin A. Bakar, F. A. Ghaleb, M. S. M. Zahid, “Path Selection With Critical Switch-Aware For Software Defined Networking”, Proceedings of the IEEE Symposium on Wireless Technology & Applications, Shah Alam, Malaysia, 17 August 2021, pp. 22-26.

[74] L. C. Freeman, “Centrality in social networks conceptual clarification”, Social Networks, Vol. 1, No. 3, 1978, pp. 215-239.

[75] L. C. Freeman, “A Set of Measures of Centrality Based on Betweenness”, Sociometry, Vol. 40, No. 1, 1977, p. 35.

[76] V. Latora, M. Marchiori, “A measure of centrality based on network efficiency”, New Journal of Physics, Vol. 9, No. 2008, 2007,

[77] B. Fan, Z. Li, N. Shu, Y. Li, “Identification of Key Nodes Based on PageRank Algorithm”, Proceedings of the IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference, Chongqing, China, 12-14 March 2021, pp. 2398-2402.

[78] A. Pekar, A. Duque-Torres, W. K. G. Seah, O. M. Caicedo Rendon, “Towards threshold-agnostic heavy-hitter classification”, International Journal of Network Management, No. 2021, pp. 1-22,

[79] C. Bi, X. Luo, T. Ye, Y. Jin, “On precision and scalability of elephant flow detection in a data centre with SDN”, Proceedings of the IEEE Globecom Workshops, Atlanta, GA, USA, 9-13 December 2013, pp. 1227-1232.

[80] M. Afaq, S. U. Rehman, W. C. Song, “A Framework for Classification and Visualization of Elephant Flow in SDN-Based Networks”, Procedia Computer Science, 2015, Vol. 65, No. Iccmit, 2021, pp. 672-681.

[81] S. Sarvotham, R. Riedi, R. Baraniuk, “Connection-level analysis and modelling of network traffic”, Proceedings of the First ACM SIGCOMM Workshop on Internet Measurement, 2001, p. 99.

[82] S. Shakkottai, N. Brownlee, K. Claffy, “A study of burstiness in TCP flow”, Lecture Notes in Computer Science, Vol. 3431, 2005, pp. 13-26, ,

[83] K. C. Lan, J. Heidemann, “A measurement study of correlations of Internet flow characteristics”, Computer Networks, Vol. 50, No. 1, 2006, pp. 46-62.

[84] A. R. Curtis, W. Kim, P. Yalagandula, “Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection”, Proceedings IEEE INFOCOM, Shanghai, China, 10-15 April 2011, pp. 1629-1637.

[85] X. Li and C. Qian, “Low-complexity multi-resource packet scheduling for network function virtualization”, Proceedings IEEE INFOCOM, Hong Kong, China, 26 April - 1 May 2015, pp.1400-1408.

[86] F. Carpio, A. Engelmann, A. Jukan, "DiffFlow: Differentiating short and long flows for load balancing in data centre networks", Proceedings of the IEEE Global Communications Conference, December 2016.

[87] F. Estrada-Solano, O. M. Caicedo, N. L. S. Da Fonseca, "NELLY: Flow Detection Using Incremental Learning at the Server Side of SDN-Based Data Centers", IEEE Transactions on Industrial Informatics, Vol. 16, No. 2, 2020, pp. 1362-1372.

[88] S. C. Madanapalli, M. Lyu, H. Kumar, H. H. Gharakheili, V. Sivaraman, "Real-time detection, isolation, and monitoring of elephant flows using commodity SDN system", Proceedings of the IEEE/IFIP Network Operations and Management Symposium, Taipei, Taiwan, 23-27 April 2018, pp. 1-5.

[89] R. Liu, H. Gu, X. Yu, X. Nian, "Distributed flow scheduling in energy-aware data centre networks", IEEE Communications Letters, Vol. 17, No. 4, 2013, pp.801-804.

[90] M. Chiesa, G. Kindler, M. Schapira, "Traffic Engineering with equal-cost-multipath: An algorithmic perspective", IEEE/ACM Transactions on Networking, Vol. 25, No. 2, 2017, pp. 779-792.

[91] S. C. Chao, K. C. J. Lin, M. S. Chen, "Flow Classification for Software-Defined Data Centers Using Stream Mining", IEEE Transactions on Services Computing, Vol. 12, No. 1, 2019, pp. 105-116.

[92] C. L. Hu, C. Y. Hsu, S. E. Khuukhenbaatar, Y. Dashdorj, Y. Dong, "Path Selection with Joint Latency and Packet Loss for Edge Computing in SDN", Proceedings of the 20th Asia-Pacific Network Operations and Management Symposium, Matsue, Japan, 18-20 September 2019.

[93] A. Alnajim and S. Salehi, "Incremental Path-Selection and Scheduling for Time-Sensitive Networks", Proceedings of the IEEE Global Communications Conference, Waikoloa, HI, USA, 9-13 December 2019.

[94] K. Gao, C. Xu, J. Qin, S. Yang, L. Zhong, G. M. Muntean, "QoS-driven Path Selection for MPTCP: A Scalable SDN-assisted Approach", Proceedings of the IEEE Wireless Communications and Networking Conference, Marrakesh, Morocco, 15-18 April 2019,

[95] N. Saha, S. Member, S. Bera, S. Member, "Sway : Traffic-Aware QoS Routing", IEEE Transactions on Emerging Topics in Computing, 2018. (in press)

[96] L. R. Prete, A. A. Shinoda, C. M. Schweitzer, R. L. S. De Oliveira, "Simulation in an SDN network scenario using the POX Controller", Proceedings of the IEEE Colombian Conference on Communications and Computing, Bogota, Colombia, 4-6 June 2014,

[97] A. Botta, A. Dainotti, A. Pescapé, "A tool for the generation of realistic network workload for emerging networking scenarios", Computer Networks, Vol. 56, No. 15, 2012, pp. 3531-3547.

[98] C. Perner and G. Carle, "Comparison of Optimization Goals for Resilient Routing", Proceedings of the IEEE International Conference on Communications Workshops, May 2019, pp. 1-6.

[99] R. Khalili, Z. Despotovic, A. Hecker, "Flow Setup Latency in SDN Networks", IEEE Journal on Selected Areas in Communications, Vol. 36, No. 12, 2018, pp. 2631-2639.

[100] H. K. Ravuri, M. T. Vega, T. Wauters, B. Da, A. Clemm, F. De Turck, "An Experimental Evaluation of Flow Setup Latency in Distributed Software Defined Networks", Proceedings of the IEEE Conference on Network Softwarization, June 2021, pp. 432-437.

[101] L. V. Morales, A. F. Murillo, S. J. Rueda, "Extending the floodlight controller", Proceedings of the IEEE 14th International Symposium on Network Computing and Applications, Cambridge, MA, USA, 28-30 September, pp. 126-133.

[102] P. Berde et al., " ONOS: towards an open, distributed SDN OS", Proceedings of the third workshop on Hot topics in software defined networking, August2014, pp. 1-6.

[103] S. Hassas Yeganeh, Y. Ganjali, " Kandoo: a framework for efficient and scalable offloading of control applications", Proceedings of the first workshop on Hot topics in software defined networks, August 2012, p. 19.

[104] N. Noorani and S. A. H. Seno, "Routing in VANETs based on intersection using SDN and fog computing", Proceedings of the 8th International Conference on Computer and Knowledge Engineering, Mashhad, Iran, 25-26 October 2018, pp. 339-344.

[105] A. T. Albu-Salih, S. A. H. Seno, S. J. Mohammed, “Dynamic routing method over hybrid SDN for flying ad hoc networks”, Baghdad Science Journal, Vol. 15, No. 3, 2018, pp. 361-368.

[106] Z. Jia, Y. Sun, Q. Liu, S. Dai, C. Liu, “CRetor: An SDN-Based Routing Scheme for Data Centers with Regular Topologies”, IEEE Access, Vol. 8, 2020, pp. 116866-116880.

[107] ONF, "Open Networking Foundation", www.opennetworking.org/mininet/ (accessed: 2022)

[108] K. Fall, K. Varadhan, The network simulator (ns-2), www. isi. edu/nsnam/ns (accessed 2022)

[109] A. Alidadi, S. Arab, T. Askari, “A novel optimized routing algorithm for QoS traffic engineering in SDN-based mobile networks”, ICT Express, Vol. 8, No. 1, 2022, pp. 130-134.

[110] P. Megyesi, A. Botta, G. Aceto, A. Pescapè, S. Molnár, “Available bandwidth measurement in software-defined networks”, Proceedings of the 31st Annual ACM Symposium on Applied Computing, April 2016, pp. 651-657,

[111] D. L. C. Dutra, M. Bagaa, T. Taleb, K. Samdanis, “Ensuring End-to-End QoS Based on Multi-Paths Routing Using SDN Technology”, Proceedings of the IEEE Global Communications Conference, Singapore, 4-8 December 2017, pp. 1-6, ,

[112] M. R. Celenlioglu, H. A. Mantar, “An SDN based intra-domain routing and resource management model”, Proceedings of the IEEE International Conference on Cloud Engineering, Tempe, AZ, USA, 9-13 March 2015, pp. 347-352.

[113] C. Huang, C. Nakasan, K. Ichikawa, H. Iida, “A multipath controller for accelerating GridFTP transfer over SDN”, Proceedings of the IEEE 11th International Conference on e-Science, Munich, Germany, 31 August - 04 September 2015, pp. 439-447.

[114] C. Huang, C. Nakasan, K. Ichikawa, H. Iida, “An SDN-Based Multipath GridFTP for High-Speed Data Transfer”, Proceedings of the IEEE 36th International Conference on Distributed Computing Systems, Nara, Japan, 27-30 June 2016, pp. 763-764.

[115] S. Tariq, M. Bassiouni, “QAMO-SDN: QoS aware Multipath TCP for software-defined optical networks”, Proceedings of the 12th Annual IEEE Consumer Communications and Networking Conference, Las Vegas, NV, USA, 9-12 January 2015, pp. 485-491.

[116] S. Tariq, M. Bassiouni, “Performance evaluation of MPTCP over optical burst switching in data centres”, Proceedings of the International Telecommunications Symposium, Sao Paulo, Brazil, 17-20 August 2014, pp. 0-4.

[117] A. Basit, S. Qaisar, S. H. Rasool, M. Ali, “SDN Orchestration for Next Generation Inter-Networking: A Multipath Forwarding Approach”, IEEE Access, Vol. 5, 2017, pp. 13077-13089.

[118] D. Eppstein, “Finding the k Shortest Paths”, SIAM Journal on Computing, Vol. 28, No. 2, 1998, pp. 652-673.

[119] S. A. Hussain, S. Akbar, I. Raza, “A dynamic multipath scheduling protocol (DMSP) for full performance isolation of links in software-defined networking (SDN)”, Proceedings of the 2nd Workshop on Recent Trends in Telecommunications Research, Palmerston North, New Zealand, 10 February 2017,

[120] N. Farrugia, V. Buttigieg, J. A. Briffa, “A globally optimised multipath routing algorithm using SDN”, Proceedings of the 21st Conference on Innovation in Clouds, Internet and Networks and Workshops, Paris, France, 19-22 February 2018, pp. 1-8.

[121] S. A. Astaneh, S. Shah Heydari, “Optimization of SDN Flow Operations in Multi-Failure Restoration Scenarios”, IEEE Transactions on Network and Service Management, Vol. 13, No. 3, 2016, pp. 421-432.

[122] S. Astaneh, S. S. Heydari, “Multi-failure restoration with minimal flow operations in software-defined networks”, Proceedings of the 11th International Conference on the Design of Reliable Communication Networks, Kansas City, MO, USA, 24-27 March 2015, pp. 263-266.

[123] A. Malik, B. Aziz, M. Bader-El-Den, “Finding most reliable paths for software-defined networks”, 13th International Wireless Communications and Mobile Computing Conference, June 2017, pp. 1309-1314,

[124] A. Malik, R. de Fréin, B. Aziz, “Rapid restoration techniques for software-defined networks”, Applied Sciences, Vol. 10, No. 10, 2020.

[125] C. Yu, Z. Zhao, Y. Zhou, H. Zhang, “Intelligent Optimizing Scheme for Load Balancing in Software

Defined Networks", Proceedings of the IEEE 85th Vehicular Technology Conference, Sydney, NSW, Australia, 4-7 June 2017, pp. 5-9.

[126] C. J. C. H. Watkins and P. Dayan, "Technical Note: Q-Leaning", Reinforcement Learning, Vol. 292, Boston, Springer, 1992, pp. 55-68.

[127] R. Miikkulainen et al. "Evolving deep neural networks", Artificial Intelligence in the Age of Neural Networks and Brain Computing, 2019, pp. 293-312.

[128] K. Gotani et al. "Design of an SDN Control Method Considering the Path Switching Time under Disaster Situations", Proceedings of the 5th International Conference on Information and Communication Technologies for Disaster Management, December 2018, pp. 1-4.

[129] K. Gotani, H. Takahira, M. Hata, L. Guillen, S. Izumi, T. Abe, "OpenFlow Based Information Flow Control Considering Route Switching Cost", Proceedings of the IEEE 43rd Annual Computer Software and Applications Conference, July 2019, Vol. 2, pp. 527-530.

[130] B. Zhao, J. Zhao, X. Wang, T. Wolf, "RuleTailor: Optimizing Flow Table Updates in OpenFlow Switches with Rule Transformations", IEEE Transactions on Network and Service Management, Vol. 16, No. 4, 2019, pp. 1581-1594.

[131] C. Wang, H. Y. Youn, "Entry aggregation and early match using hidden Markov model of flow table in SDN", Sensors, Vol. 19, No. 10, 2019.

[132] W. K. Jia, X. Wang, "Flow aggregation for large-scale SDNs with scattered address space allocation", Journal of Network and Computer Applications, Vol. 169, 2020, p. 102787.

[133] Y. Guo, H. Luo, Z. Wang, X. Yin, J. Wu, "Routing optimization with path cardinality constraints in a hybrid SDN", Computer Communications, Vol. 165, 2021, pp. 112-121.

[134] R. Maaloul, R. Taktak, L. Chaari, B. Cousin, "Energy-Aware Routing in Carrier-Grade Ethernet Using SDN Approach", IEEE Transactions on Green Communications and Networking, Vol. 2, No. 3, 2018, pp. 844-858.

[135] S. Civanlar, M. Parlakisik, A. M. Tekalp, B. Gorkemli, B. Kaytaz, E. Onem, "A QoS-enabled OpenFlow environment for scalable video streaming", Proceedings of the IEEE Globecom Workshops, Miami, FL, USA, 6-10 December 2010, pp. 351-356.

[136] H. Owens and A. Durresi, "Video over Software-Defined Networking (VSDN)", Computer Networks, Vol. 92, 2015, pp. 341-356.

[137] T. F. Yu, K. Wang, Y. H. Hsu, "Adaptive routing for video streaming with QoS support over SDN networks", Proceedings of the International Conference on Information Networking, Cambodia, 12-14 January 2015, pp. 318-323.

[138] S. Yilmaz, A. M. Tekalp, B. D. Unluturk, "Video streaming over software-defined networks with server load balancing", Proceedings of the International Conference on Computing, Networking and Communications, Garden Grove, CA, USA, 16-19 February 2015, pp. 722-726.

[139] A. Chooprateep, Y. Somchit, "Video Path Selection for Traffic Engineering in SDN", Proceedings of the 11th International Conference on Information Technology and Electrical Engineering, Pattaya, Thailand, 10-11 October 2019.

[140] A. Jüttner, B. Szviatovszki, I. Mécs, Z. Rajkó, "Lagrange relaxation based method for the QoS routing problem", Proceedings of the Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society, Anchorage, AK, USA, 22-26 April 2001, pp. 859-868.

[141] D. Kotani, Y. Okabe, "A Packet-In message filtering mechanism for protection of control plane in OpenFlow switches", Proceedings of the ACM/IEEE Symposium on Architectures for Networking and Communications Systems, Marina del Rey, CA, USA, 20-21 October 2014, pp. 695-707.

[142] J. Yan, H. Zhang, Q. Shuai, B. Liu, X. Guo, "HiQoS: An SDN-based multipath QoS solution", China Communications, Vol. 12, No. 5, 2015, pp. 123-133,

[143] S. Tomovic, N. Prasad, I. Radusinovic, "SDN control framework for QoS provisioning", Proceedings of the 22nd Telecommunications Forum Telfor, Belgrade, Serbia, 25-27 November 2014, pp. 111-114,

[144] B. G. Assefa, O. Ozkasap, "MER-SDN: Machine learning framework for traffic-aware energy-efficient routing in SDN", Proceedings of the IEEE 16th International Conference on Dependable, Auto-

nomic and Secure Computing, 16th International Conference on Pervasive Intelligence and Computing, 4th International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress, 2018, pp. 966-973.

[145] G. C. Deng, K. Wang, "An Application-aware QoS Routing Algorithm for SDN-based IoT Networking", Proceedings of the IEEE Symposium on Computers and Communications, Natal, Brazil, 25-28 June 2018, pp. 186-191.

[146] C. X. Cui, Y. B. Xu, "Research on load balance method in SDN", International Journal of Grid and Distributed Computing, Vol. 9, No. 1, 2016, pp. 25-36.

[147] O. Ahmed, F. Ren, A. Hawbani, Y. Al-Sharabi, "Energy Optimized Congestion Control-Based Temperature Aware Routing Algorithm for Software Defined Wireless Body Area Networks", IEEE Access, Vol. 8, 2020, pp. 41085-41099.

[148] L. El-Garoui, S. Pierre, S. Chamberland, "A new sdn-based routing protocol for improving delay in smart city environments", Smart Cities, Vol. 3, No. 3, 2020, pp. 1004-1021.

[149] Y.-S. Yu and C.-H. Ke, "Genetic algorithm-based routing method for enhanced video delivery over software-defined networks", International Journal of Communication Systems, Vol. 31, No. 1, 2018, p. 3391.

[150] S. Abbas, N. Javaid, A. Almogren, S. M. Gulfam, A. Ahmed, A. Radwan, "Securing Genetic Algorithm Enabled SDN Routing for Blockchain-Based Internet of Things", IEEE Access, Vol. 9, 2021, pp. 139739-139754,

[151] D. Li, X. Wang, Y. Jin, H. Liu, "Research on QoS routing method based on NSGAII in SDN", Journal of Physics: Conference Series, Vol. 1656, No. 1, 2020,

[152] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, G. Srivastava, S. Mohan, A. M. Rababah, "Cost optimization of secure routing with untrusted devices in software-defined networking", Journal of Parallel and Distributed Computing, Vol. 143, 2020, pp. 36-46.

[153] E. F. Morales, J. H. Zaragoza, "An introduction to reinforcement learning", Decis. Theory Model. Appl. Artif. Intell. Concepts Solut., 2011, pp. 63-80.

[154] J. Jiang, L. Hu, P. Hao, R. Sun, J. Hu, H. Li, "Q-FDBA: Improving QoE fairness for video streaming", Multimedia Tools and Applications, Vol. 77, No. 9, 2018, pp. 10787-10806,

[155] Z. Yuan, P. Zhou, S. Wang, X. Zhang, "Research on Routing Optimization of SDN Network Using Reinforcement Learning Method", Proceedings of the 2nd International Conference on Safety Produce Informatization, Chongqing, China, 28-30 November 2019, pp. 442-445.

[156] C. Fang, C. Cheng, Z. Tang, C. Li, "Research on Routing Algorithm Based on Reinforcement Learning in SDN", Journal of Physics: Conference Series, Vol. 1284, No. 1, 2019,

[157] T. T. Huong, N. D. D. Khoa, N. X. Dung, N. H. Thanh, "A global multipath load-balanced routing algorithm based on Reinforcement Learning in SDN", Proceedings of the International Conference on Information and Communication Technology Convergence, Jeju, Korea, 16-18 October 2019, pp. 1336-1341.

[158] B. G. Assefa and O. Ozkasap, "HyMER: A Hybrid Machine Learning Framework for Energy Efficient Routing in SDN", Proceedings of the IEEE 16th International Conference on Dependable, Autonomic and Secure Computing, 16th International Conference on Pervasive Intelligence and Computing, 4th International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress, August 2018, pp. 974-980.

[159] Z. Shi, J. Zhu, H. Wei, "SARSA-based delay-aware route selection for SDN-enabled wireless-PLC power distribution IoT", Alexandria Engineering Journal, Vol. 61, No. 8, 2022, pp. 5795-5803.

[160] A. Ouhab, T. Abreu, H. Slimani, A. Mellouk, "Energy-efficient clustering and routing algorithm for large-scale SDN-based IoT monitoring", Proceedings of the IEEE International Conference on Communications, Dublin, Ireland, 7-11 June 2020.

[161] C. Yu, J. Lan, Z. Guo, Y. Hu, "DROM: Optimizing the Routing in Software-Defined Networks with Deep Reinforcement Learning", IEEE Access, Vol. 6, 2018, pp. 64533-64539,

[162] W. X. Liu, "Intelligent Routing based on Deep Reinforcement Learning in Software-Defined Data-Center Networks", Proceedings of the IEEE Symposium on Computers and Communications, Barcelona, Spain, 29 June - 3 July 2019.

[163] W. Liu, J. Cai, Q. C. Chen, Y. Wang, “DRL-R: Deep reinforcement learning approach for intelligent routing in software-defined data-center networks”, Journal of Network and Computer Applications, 2020, p. 102865.

[164] D. N. Maheswari, C. B. Sujitha, “Routing Optimization in Sdn Using Deep Reinforcement Learning”, Journal of Engineering, Computing and Architecture, Vol. 10, No. 5, 2020, pp. 40-47.

[165] C. Xu, W. Zhuang, H. Zhang, “A Deep-reinforcement Learning Approach for SDN Routing Optimization”, Proceedings of the 4th International Conference on Computer Science and Application Engineering, October 2020.

[166] Y. R. Chen, A. Rezapour, W. G. Tzeng, S. C. Tsai, “RL-Routing: An SDN Routing Algorithm Based on Deep Reinforcement Learning”, IEEE Transactions on Network Science and Engineering, Vol. 7, No. 4, 2020, pp. 3185-3199.

[167] X. Guo, H. Lin, Z. Li, M. Peng, “Deep-Reinforcement-Learning-Based QoS-Aware Secure Routing for SDN-IoT”, IEEE Internet of Things Journal, Vol. 7, No. 7, 2020, pp. 6242-6251.

[168] W. Qi, Q. Song, X. Kong, L. Guo, “A traffic-differentiated routing algorithm in Flying Ad Hoc Sensor Networks with SDN cluster controllers”, Journal of the Franklin Institute, Vol. 356, No. 2, 2019, pp. 766-790.

[169] S. Torkzadeh, H. Soltanizadeh, A. A. Orouji, “Energy-aware routing considering load balancing for SDN: a minimum graph-based Ant Colony Optimization”, Cluster Computing, Vol. 24, No. 3, 2021, pp. 2293-2312.

[170] M. Hayes, B. Ng, A. Pekar, W. K. G. Seah, “Scalable Architecture for SDN Traffic Classification”, IEEE Systems Journal, Vol. 12, No. 4, 2018, pp. 3203-3214.

[171] A. M. Abdelmoniem, B. Bensaou, A. J. Abu, “Mitigating incast-TCP congestion in data centres with SDN”, Annals of Telecommunications, Vol. 73, No. 3-4, 2018, pp. 263-277.

[172] A. Gutnikov, "DDoS attacks in Q4 2021", www.securelist.com/category/ddos-reports/ (accessed: 2022)

[173] B. Isyaku, K. Bin Abu Bakar, M. N. Yusuf, M. S. Mohd Zahid, “Software Defined Networking Failure Recovery with Flow Table Aware and flows classification”, Proceedings of the 11th IEEE Symposium on Computer Applications & Industrial Electronics, April 2021, pp. 337-342.