A Study on Reinforcement Learning-Based Traffic Engineering in Software-Defined Networks

A. Bhavani, Y. Ekshitha, A. Mounika, and U. Prabu

Abstract Modern communication networks have grown highly complex and dynamic, making them difficult to describe, forecast, and govern. So, the software-defined networks (SDNs) have emerged. It is a centralized network, and it is flexible to route network flows. Traffic engineering (TE) technologies are used with deep reinforcement learning (RL) in SDN to make networks more agile. Different strategies for network balance, improvement, and minimizing maximum link usage in the overall network were considered. In this article, recent work on routing as well as TE in SDN and hybrid SDN is analyzed. The mathematical model and algorithm used in each method are interpreted, and an in-depth analysis has been done.

Keywords Software-defined networks · Hybrid software-defined networks · Traffic engineering · Routing · Reinforcement learning

1 Introduction

SDN aims to separate the control plane, and data plane to make network management easier, cut operating costs, and encourage innovation and development [1]. It is a networking design that allows for a dynamically, efficient network arrangement to boost overall network ability while also making networks more agile and adaptable [2]. It enables software applications to govern the network centrally. This allows operators to control the whole network. It allows a controller to make centralized decisions and adaptively design packet-switching nodes [3]. The SDN architecture is shown in Fig. 1.

A. Bhavani · Y. Ekshitha · A. Mounika · U. Prabu (✉)
Department of Computer Science and Engineering, Velagapudi Ramakrishna Siddhartha Engineering College, Vijaywada, India
e-mail: uprabu28@gmail.com

S. Smys et al. (eds.), *Computer Networks and Inventive Communication Technologies*, Lecture Notes on Data Engineering and Communications Technologies 141,
https://doi.org/10.1007/978-981-19-3035-5_4

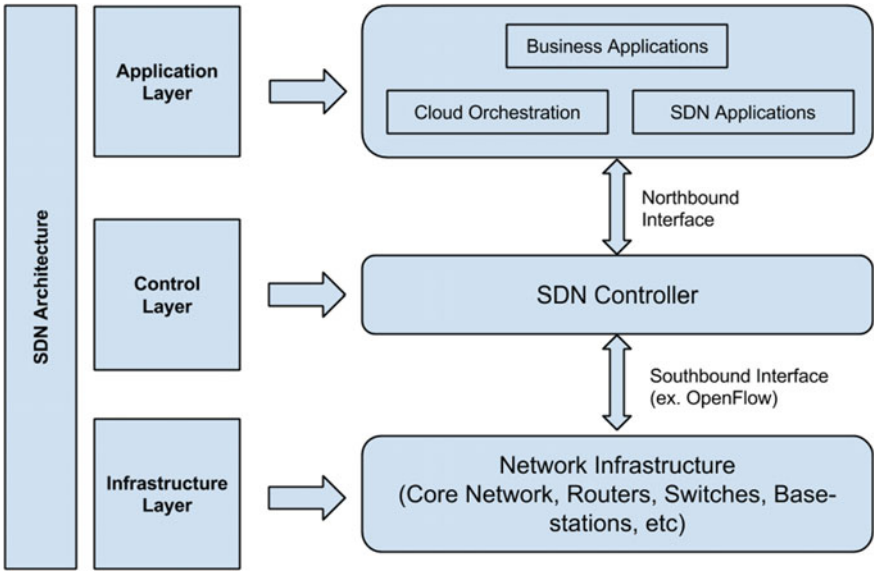

Fig. 1 SDN architecture [4]

1.1 Hybrid SDN

A hybrid SDN may be a networking strategy that mixes ancient networking with software-defined network manner within the same atmosphere [5]. Engineers will operate SDN technologies and old shift rules on constant physical hardware in a very hybrid SDN [3]. Whereas traditional distributed networking protocols still steer the bulk of the traffic on the network, a network manager will style the SDN management plane to seek out and govern sure traffic flows.

RL-Routing is a reinforcement learning routing technique that solves a traffic engineering (TE) problem in terms of throughput and delay in an SDN. Instead of constructing an exact mathematical model, RL-Routing tackles the TE problem through experience. One-to-many network setup for routing choices is used and extensive network information for state representation is considered. The reward mechanism, which uses network throughput and delay to optimize network throughput, can be adjusted to optimize upward or downward network throughput. The agent develops a policy that anticipates future behavior of the underlying network and offers improved routing paths between switches after receiving suitable training.

1.2 Traffic Engineering

TE is a network management technology that enhances the performance of networks by optimizing traffic routing approaches and communicating by anticipating and controlling the behavior of transmitted data [1]. Using data flow to determine link status paths within a network balances the load on multiple connections, routers, and switches [6]. This is particularly important in networks with several parallel paths.

1.3 Routing

Routing is the method of choosing the most convenient path through the associate network to send packets to a destination host or hosts so that the router forwards the packets to those hosts [6]. It is the method of creating a traffic path among a network, likewise as across and across many networks. The neural network's purpose is to reduce network time delay while optimizing the packet pathways that are being addressed. The shortest path is regarded the most important issue in any routing method that may be carried out in real time.

1.4 Reinforcement Learning

RL could be an ML coaching strategy that rewards fascinating behaviors whereas laborious undesirable ones [7]. A reinforcement learning agent will understand and comprehend its surroundings, act, and learn through trial and error usually. It is all regarding working out a way to behave optimally during a given scenario to maximize reward. Associate degree agent explores associate degree unknown atmosphere to attain a goal within the reinforcement learning downside [8]. RL is made on the concept that the increasing of expected accumulative reward could also be wont to represent any goal. To maximize reward, the agent should learn to sense and disturb the state of the atmosphere through its activities.

Both deep learning and reinforcement learning are self-learning systems. Deep learning involves learning from a training set and then applies that knowledge to fresh data, whereas reinforcement learning involves dynamically learning by altering actions depending on continuous response to enlarge a reward. Reinforcement learning and deep learning are not mutually inconsistent.

1.5 Deep Reinforcement Learning

Deep reinforcement learning may be a variety in ML that permits intelligent robots to find out from their actions within the same from that folk do [9]. The fact that associate agent is rewarded or penalized to support their actions is inherent during this variety of machine learning [10]. Deep RL may be an answer that features deep learning permitting agents to form selections supported by unstructured computer files while not manually constructing the state area.

This paper's primary contributions are summarized as follows:

- The article shows different approaches to improve network performance using TE strategies with reinforcement learning in SDN and hybrid SDN.
- Various routing methods in the software-defined network to improve network efficiency.
- The paper shows a comparison of various traffic engineering and routing algorithms in SDN and hybrid SDN.

The remainder of this paper is organized as follows: Sect. 2 represents the various TE methods in SDN and hybrid SDN using deep RL. Section 3 describes the different routing approaches in SDN, and Sect. 4 represents the comparison of various traffic engineering and routing algorithms in SDN and hybrid SDN.

2 TE in SDN and Hybrid SDN Using Reinforcement Learning

2.1 TE in Hybrid SDN

Guo et al. [6] proposed a novel method on the TE in hybrid SDN. With the rise of software-defined networks (SDNs), network routing is becoming more centralized and flexible. Traffic engineering in hybrid SDNs is a topic that is attracting wide interest from academia and industry. RL-based traffic splitting method that learns to balance the dynamically changing traffic and address it through a traffic splitting agent in hybrid SDN. To generate a routing strategy for new traffic needs quickly and intelligently, a traffic splitting agent is created and learned offline using the RL algorithm to build a direct link between traffic demands and traffic splitting rules. The powerful traffic splitting guidelines provides the policies that can be used to set up the traffic splitting ratios on SDN switches. These switches can be advanced quickly and can expand rapidly once the traffic splitting agent has been learned [11]. It is usually recommended to create a suitable simulation environment to avoid routing loops. The traffic splitting guidelines are used to fulfill the interactive standards.

An RL approach for tackling the TE problem of the hybrid SDN consists of two steps: offline learning and online routing. Draw the directed acyclic graph (DAG) [12] based on hybrid SDN architecture and traffic statistics to ensure a hybrid SDN

environment with loop-free routing for the RL agent's interaction during the offline learning stage. A traffic-partitioning agent is taught to make a direct link between the network environment and the routing techniques using the RL methods and the constructed DAG. When the demand of the traffic changes, the trained traffic splitting agent can quickly establish an acceptable routing scheme at the online routing stage.

The TE problem's network model is a hybrid SDN with dynamic traffics. The network topology is represented as an undirected graph $H = (X, F)$, where X is the set of forwarding devices, which is made up of SDN switch Xs and router $X1$, and F is the link set, with $C(f)$ denoting the capacity of connection $f \in F$.

The group of (TMs) may be indicated by $b = \{B_1, B_2, \ldots, B_n\}$ also r_i may be the weight constant of TM b_i. Those components $B_i(p, q)$ to b_i speak the traffic demand starting from gadget p to gadget $q.p,q \in X$, W is the link weight setup under the OSPF protocol. Variable e means the partitioning of traffic flows.

A collection of TMs B, the purpose of TE on the hybrid SDN is to increase the network performance by minimizing the maximum link usage (MLU) [13] for every single TM B_i. In the hybrid SDN, both the link weight settings w and the partitioning traffic movement f on each SDN switch decide the result of MLU. TE problems on hybrid SDN are defined as

$$\textbf{Minimize} \sum_{i=1}^{n} r U_i^{\max}$$

$$\sum_{j=1}^{n} r_i = 1, \quad 0 \le r_j \le 1$$

$$w(f) \in N, \ \forall f \in F$$

$$0 \le U_i^{\max} \le 1$$

A node $q \in X$ and TM $B_i \in B$, we calculate node q's MLK as below:

$$\text{MLK } B(v) = \max U_i(e)$$

The value of MLK of node q in different TMs is calculated based on MLK(q) by computing equation

$$\text{score}(v) = \sum_{B_i \in B} \text{MLK } B_i(v)$$

The TE's goal in hybrid S reduces the MLU as much as possible. As a result, MLU is included in constructing reward r_t as defined to enable the traffic-partitioning agent to effectively learn optimal rules with small MLU

$$r_t = \begin{cases} -e^{2(\frac{1}{a}-1)} \\ 0 \\ e^{2(a-1)} \end{cases}$$

$$a = U_i^{\max} / U_t^{\max}$$

The index a represents the level of performance improvement after traffic-partitioning rules are implemented at t time.

RL approach is compared to other methods such as open shortest path first and WA-SRT. RL technique outperforms the OSPF method in terms of MLU reduction and comes close to the WA-SRTE method [14]. The RL approach demonstrates the ability to relate network status to traffic-partitioning rules, assisting in the link capacity balancing process.

2.2 TE in Hybrid SDN

Zhang et al. [5] proposed a novel method for TE issues with RL. By rerouting as several flows as potential, traffic engineering approaches area units capable of achieving the best performance [13]. One TE methodology for mitigating the impact of a network outage is to balance link usage by forwarding the bulk of traffic matrices that area unit wedged. The networking unit provides two examples: ECMP [15] and resending form of essential flows utilizing a code package. Extremely, crucial flows rerouting RL is another RL technique, mechanically learn the rules for choosing needed flow for every TM. CFR-RL formulates and solves an easy arithmetic drawback so as to direct these essential flows so as to balance network association usage.

An RL-based technique for balancing network link utilization by learning critical flows selecting strategy and rerouting relevant essential flow. CFR-RL uses the reinforce [16] technique to train this neural network, with minor tweaks. Reward and state space of a key flow selection strategy utilizing a customized RL technique are all included. State space inputs are the traffic matrix, and RL issue necessitates a huge action space containing massive network nodes, with the LP function as a reward. ECMP routing is used to disperse the traffic.

State: An agent is given state $c_t = T_t$, where T_t is TM at time t that provides information about each flow's traffic demand.

Action Space: Action spaces are of two types. They are discrete action space and continuous action space. Here, the agent chooses from a finite action set which distinct action to do using a discrete action space. Actions are conveyed as a single real-valued vector in a continuous action space. CFR-RL would choose P essential flows for each state st. Given that a network with K nodes has a total of $K * (K - 1)$ flows, RL issue will necessitate huge action space the value of the $C^p_{k*(k-1)}$. Permit agent to some P distinct actions in every time t by setting the action space to $(0, 1, \ldots, (K * (K - 1)))$ [17].

Reward: CFR-RL resends these key flows and gets the highest link usage U by resolving the problem of routing optimization after sampling P distinct essential flows for state c_t. Reward q is set to $1/U$ that reflects network performance after crucial traffic is rerouted to balance link use.

The following is a description of the crucial flow rerouting problem. The goal is to acquire optimal routing ratios $\sigma_{s,d}^{i,j}$ for each essential flow, so in MLU, U is minimized, given a network $H(X, F)$ with the group of demands of traffic Bs, d for the set of essential flows (fk) and link load $L_{i,j}$ supplied by the rest of the flows that are utilizing the default settings. We construct the problem of rerouting as an optimization to find all viable under-utilized pathways for the specified important flows

$$\textbf{minimize } U + \sum_{(i,j)\in F} \sum_{(s,d)\in f_k} \sigma_{i,j}^{s,d}$$

The optimal routing result for chosen important flow is derived by addressing the aforementioned LP problem with LP solvers. The SDN controller then installs and changes flow entries at the switches in the appropriate order.

The CFR-RL scheme is presented with the goal of lowering maximum link utilization in a network and minimizing network disturbance that causes service disruption. By resending just 11–20.3% of the entire traffic, CFR-RL delivers near-optimal performance. CFR-RL achieves optimal load balancing performance in excess of 95% of the time.

2.3 ScaleDRL Scheme for TE in SDN Using Pinning Control

Sun et al. [10] proposed a method for SDN issues using TE. Deep reinforcement learning and software-defined learning able to develop a model-free TE system with the assistance of networking technologies. Existing DRL-based TE results, on the opposite hand, all have a quantifiability issue that forestalls them from getting used in giant networks. A method that mixes management theory associate degreed DRL technology is planned to develop an economical network management approach for TE [18]. ScaleDRL could be a planned approach that employs the construct of promise management theory to spot and label the group of network links as essential. Supported traffic distribution info received by SDN controller, DRL approach is utilized to effectively change the collection of link loads for vital links. The forwarding pathways of network flows are often dynamically modified employing a weighted shortest path technique.

ScaleDRL presents a mechanism for evaluating the importance of network links in routing path development. A control theory-based flow selection algorithms is developed on this assessment approach [19]. ScaleDRL adapts the DRL method to manage communication network traffic allotment statistics and dynamically construct TE

regulations. ScaleDRL is validated using OMNet++ in a fine-grained simulation with several network topologies of various sizes.

The link weights are the goal of DRL algorithm in ScaleDRL. As a result, in order to pick a fraction of data plane links with pinning control, we basically introduce the concept of equilibrium to define the flow relevance in overall network's routing patterns. Link centrality, in particular, refers to the correlations that exist between links as a result of routing pathways.

For network H, R is to represent the group of node r and F is to represent links f between nodes, having $H = (R, F)$. For a couple of node r_i, $r_j \in R$, that shall contain at least a single readdressing path $p_{i,j} = \{f1, f2, \ldots, f|o|\}$ that will send traffic from r_i to r_j nodes, and represent the shortest path between them as $o^*_{i,j}$.

With weighted shortest path algorithms, computed the shortest path on H, where value of weight for the link fm is represented by w_m $(0 \leq m \leq |F|)$. Use the indicator $y^m_{i,j}$ to denote if link fm is acquire in $o^*_{i,j}$ that is $y^m_{i,j} = 1$ if $o^*_{i,j}$ carry fm, $y^m_{i,j} = 0$. The equilibrium of the link fm is defined as

$$f_m = \frac{\sum_{i=1}^{|R|} \sum_{j=1}^{|R|} x^m_{i,j}}{|R| \times |R|}$$

A MDP is used to define model DRL's working process (MDP) [20]. The DRL algorithms in the MDP interconnect with target environment. MDP is characterized as follows:

$$N = (T, A, Q, S, Z),$$

with T represents state space s, A represents action space a, Q represents reward space q, S represents the group of probability P, and Z denoting a discount factor.

The entire training goal of DRL algorithm is to increase the cumulative rewards

$$Q = \sum_{t=0}^{T} \gamma^t q_t$$

In numerous network topologies, the packets simulation reveals the ScaleDRL decreases typical end coordinated universal time up to 40% in comparison with progressive DRL-based TE system. The link centrality-based choice theme had the best performance of all the schemes studied that confirms the link centrality-based choice strategy.

2.4 RL Approach for TE Based on Link Control

Xu et al. [3] proposed a method for TE issues using RL. Deep reinforcement learning (DRL) permits to use of machine learning to make a model-free TE theme.

Existing DRL-based TE solutions, on the opposite hand, cannot be utilized in massive networks. To develop a TE theme, a theme that mixes management theory and DRL is bestowed. The prompt arrange ScaleDRL selects link range in network and names the essential flows employing a construct from promise management theory [21]. A DRL technique is employed to dynamically alter the weights of links for essential flows supported traffic distribution statistics. The forwarding pathways of the flows will be dynamically changed employing a weighted shortest path technique.

The basic idea for pinning management is during advanced and dominant whole network elements to attain network state consumes a lot of resources within the management algorithmic rule and so does not happen; instead, bound management signals in mere a part of the network may be removed to attain the supposed synchronization mode. ScaleDRL is enforced supported SDN. There is a vital link algorithmic rule and a DRL algorithmic rule [22] that resides within the ScaleDRL management. With SDN, traffic distribution may be collected from time to time by the controller, and TE rules may be upgraded sporadically. Supported this technology, ScaleDRL operates in 2 categories: offline and online. Within the offline section, the link choice algorithmic rule analyzes constellation and a select group of flows as important network flows supported pin-control perspective. Within the online section, the DRL algorithmic rule manages flow weights to direct traffic networks.

In link weight algorithm [19], for network H, R is to represent the group of node r, f is to represent links f between nodes, having $H = (R, F)$. For a couple of nodes $r_i, r_j \in R$, that shall contain at least a single readdressing path $p_{i,j} = \{f1, f2, \ldots, f|o|\}$ that will send traffic from r_i to r_j nodes, and represent shortest path between them as $o^*_{i,j}$.

With weighted shortest path algorithms, computed the shortest path on H, where value of weight for the link f_m is represented by w_m $(0 \leq m \leq |\mathrm{F}|)$. Use the indicator $y^m_{i,j}$ to denote if link f_m is acquire in $o^*_{i,j}$ that is $y^m_{i,j} = 1$ if $o^*_{i,j}$ carry f_m, $y^m_{i,j} = 0$. The equilibrium β of the link f_m is defined as

$$\beta(b_m) = \frac{\sum_{i=1}^{|R|} \sum_{j=1}^{|R|} y^m_{i,j}}{|R| \times |R|}$$

DRL algorithm develop an action with in neural network to the surroundings at time t of MDP supported the observation of the state k_t. The DRL algorithm obtains reward r_t after a_t is performed in the environment, which assesses a_t's performance in the environment. There are currently several forms of DRL algorithms, with the key distinction being the mechanism used to update the neural network parameters in DRL. The DRL framework we are using is ACKTR [23].

The packets simulation reveals the ScaleDRL decreases typical end coordinated universal time up to 40% in comparison with progressive DRL-based TE system.

3 Routing in SDN Using Deep Reinforcement Learning

3.1 RL-Routing: An SDN Routing Algorithm Based on DRL

Chen et al. [7] proposed a method for routing in SDN based on RL. Because communication networks have become so complex and dynamic, they are challenging to describe and anticipate. To overcome traffic engineering challenge of SDN with respect to throughput, latency to create a reinforcement learning routing method. Instead of constructing an exact mathematical model, RL-Routing tackles TE problems through training. To employ a one-to-many network setup for routing options and use extensive network information for state representation. The reward method, which will use networks work rate and delay to optimize network throughput, may be adjusted to optimize up or down network work rate. The algorithm develops a rule which anticipates further behavior of basic networks and offers improved routing pathways among switches after receiving suitable training.

The RL-Routing is located, how it interconnects with remaining components in SDN architecture. For message exchange, the controller attaches to switch over the OpenFlow path. There are two main modules in the RL-Routing application. They are network monitoring modules, which obtain network information through passive and active network measurements [24]. Network information relates to the position of network tool, such as flow delay and flow work rate. Another module is the action translator module, which converts the algorithm chosen action into the suitable group of OpenFlow messages to modify switch link tables.

Assume network as DGH(X, F) where $X = \{s_1, s_2, \ldots s_n\}$ is set of switches and $F \subseteq X \times X$ is the group of the flows in a network, $|F| = m$. Consider the network flows are two-way directional means $f_{i,j}$ and f_j, I are upward and downward flows connect to s_i. Neighbors of switches s_i are $N(s_i) = \{s_j \in X\}.F(s_j) = \{f_i, k \in F\}$ is group of adjacent edge to s_i. s_{src} is the source switch.

Let Dsrc $\subset X - \{s_{\text{src}}\}$ be collection of entire destination switch from s_{src}. Path $p_{\text{src,des}}$ is a path in network $H(X, F)$ that interconnects s_{src} to s_{des} to the order of switch $(s_{\text{src}}, s_i, s_j, s_k, s_{\text{des}})$, where the besides switches in order form edges in F, and every switch will be visited once. $b_t(f_{i,j})$ is link bandwidth $f_{i,j}$ that interconnects s_i to s_j at period intervals Δt. Delay$(f_{i,j})$ anderror $t(f_{i,j})$ represent delay of link and indicator for the fault occurred in $f_{i,j}$ for time intervals Δt. Bandwidth of path $b_t(p_{\text{src,des}}) = \min b_t(f_{i,j})$ is min bandwidth of the link at period intervals Δt. Delay of the path $\text{delay}_t(p_{\text{src,des}}) = \sum_{f_{i,j}} \text{delay}(f_{i,j})$ is addition of flow delay in path at period intervals Δt.

The problem of TE is represented as given $H(X, F)$, Ssrc, Dsrc and finds a group of flows for the succeeding ssrc's data to switch in Dsrc. The aim is to increase swsrc's work rate and reduces communications delay.

The description of RL-Routing is given here. Then, provided a Q learning algorithm [25] to resolve the traffic engineering problem.

Description of RL-Routing

1. The template for routing is designated as $O = (L, B, S, U, N)$ where $L \subset Sz$ represents state space.
2. B represents action space.
3. $S : T \times B \rightarrow S$ represents reward function.
4. U represents transition probability.
5. $N \in [0, 1]$ represents discount rate.

On various network topologies, simulation output demonstrates the RL-Routing earns greater reward and allows host to send the big file quicker than OSPF and LL algorithm. On the NSF Net topology, for example, the total of RL-rewards routings is 119.30, where OSPF and LLs are 106.59 and 74.76. The average RL-Routing transmission period for 40 GB file is 25.2 s. OSPF and LL have 63 and 53.4 s.

3.2 TIDE: Time-Relevant Deep Reinforcement Learning for Routing Optimization

Sun et al. [8] proposed a novel method for routing issues. TIDE is a smart network control architecture in view of DRL that can progressively advance routing algorithms in an SDN network without requiring human cooperation. TIDE has been completely tried and executed in a real-world network setting. The discoveries of the test show that TIDE can change the directing system powerfully founded on the network and can limit in general network sending delay by about 9% contrasted with standard calculations. The optimization has been studied for a protracted time in network style, and several other optimization ways are given by each lecturers and business. However, such systems are either too troublesome to use in applications or perform poorly. AI-based routing ways are planned in early years, with an emergence of SDN and computing. TIDE, associate degree intelligent network management design supported DRL that may effectively improve routings ways in associate degree network while not requiring intervention of human, is planned during this study. TIDE is tested and enforced during a real-world network surroundings. The results of the experiment show that TIDE will dynamically adapt the routing strategy supported the network state of affairs and improve the full network transmittal delay.

To implement the automated routing strategy in SDN, an initial have to be compelled to build an Associate in Nursing intelligent network management design known as TIDE [26]. 3 logic planes form up the recommended design for intelligent network control: information plane, management plane, and AI plane. There are 3 components to the intelligent call loop: reward, state and assortment, rule development, and policy preparation. The most contributions are to execute intelligent routing management of a sending network, "collections-decision-adjustment" loop is bestowed, and RNN-based DRL system is rigorously created for abstracting traffic properties and might effectively develop a closure optimal routing set up counting on the ever-changing traffic distribution.

The fundamental method of TIDE is DDPG [27], a DRL framework for constant control. DDPG's result is not distinct as a narrow group of some actions, unlike the bulk of reinforcement learning models like DQN. In order to elegantly regulate entire network traffic flows, routing optimization usually requires adjusting the link capacity for every link. As a result, the link capacity space value in the network should be large, making constant algorithms like DDPG is a good choice for creating routing strategies.

The interconnection process between agent and environment is viewed as a MDP in RL. The element tuple of MDP is $O = (V, B, K, D, Z)$, where V represents state space v, B represents action space b, K represents reward space k, D represents transition probability method, and $z[0, 1]$ represents discount factor. An agent selects action b under state v according to rule, which is represented as $(b|v)$ in normal rules and $b = (v)$ in deterministic rules.

Value functions are used to determine if a policy is beneficial or not. The value of C is a prominent value function in reinforcement learning. When choosing action b in state v, value of policy C is defined as [28]

$$C(st, b) = E\left[\sum_{k=0}^{\infty} y^k K(v_{t+k}, b_{t+k})\right]$$

TIDE decreases the overall transmission latency of entire traffic by around 9%. This is due to the growing unpredictability of noise traffic, which makes it more difficult for TIDE categorize network traffic, limiting TIDE's capacity to make perfect decisions.

3.3 QR-SDN: Toward Reinforcement Learning States, Actions, and Rewards for Direct Flow Routing in Software-Defined Networks

Rischke et al. [9] proposed a novel method for SDN issues. QR-SDN could be an ancient tabular reinforcement learning system that builds and evaluates routing patterns of single flows in an action statehouse. The findings are accustomed produce a model-free reinforcement learning strategy. Owing to direct illustration of link routes within QR-SDN action statehouse, QR-SDN is the initial RL-Routing technique that changes many routing ways in which among given offer switch destination try whereas holding flow integrity. In alternative words, with QR-SDN, packets from an eternal flow follow a set routing path, however, flows from a continuing source destinations switches might take a spread of routes. QR-SDN tends to be enforced in an exceedingly extremely SDN compete for the testbed.

The presentation of SDN link routing drawback to the economical higher cognitive process by RL agent is not totally been investigated. The planning of the states and actions, particularly, should be self-addressed as to adequately represent link routing

drawback for a process by RL algorithm, to see, however, with success an action solves the flow routing drawback, we tend to utilize the reward. The total of latencies on these pathways of the flows is the planned incentive.

Assume the network $H(W, X)$, where X is a collection of edges that connects a group of vertices W. We prefer to focus on a single communication flow, that is, the flows that convey data from a single sender to a single receiver. Flow e represents information transfer from a given sender se to a given receiver de for a certain application or transport layer context, such as a given TCP flow. E is commonly used to represent the group of all flows. We usually assume that flow e transfers a certain traffic rate Q_e into the network from supply host.

The path $V_{s,d}$ is an order of vertices $V = (p_1, \ldots, p_n)$ from a group of every possible path $V \in V_{s,d}$ interconnecting s to d, where group $V_{s,d}$ might be defined by search algorithms like DFS [29–33].

The SDN controller's RL agent monitors the environment by monitoring the required main performance indicator, as bandwidth, at different times $t = 0, 1, 2 \ldots$. The observation contains reward $S_t \in S \subset S$ and the environment's state R_t from the group of states $R = \{R_1, R_2, \ldots\}$.

The state R_t should be made up of a table with the presently chosen paths Q for every flow e. An action $B_t \in B$ is chosen based on the state R and its accompanying reward S. The group of alternative paths including the present path determines the set of actions $B = \{B_{t,1}, B_{t,2}, \ldots\}$. The total latencies L_e along the present routes $V_{s,d}$ of flows $e \in E$ is reward S_t.

For moderate to high loads, the link-preserving different routing of paths QR-SDN provides much lesser latencies than conventional unicast path routing systems, according to the tests. Shifts, like load changes owing to additional flows that end, are successfully accommodated by QR-SDN.

4 Comparison of Various Traffic Engineering and Routing Algorithms in SDN and Hybrid SDN

In the section, various traffic engineering techniques and routing algorithms used in SDN and hybrid SDN are analyzed and given in Table 1.

The mentioned algorithms or techniques (Table 2) are used to improve the performance of the network and to increase the efficiency of routing in software-defined networking. Deep reinforcement algorithms and traffic engineering techniques that mentioned in the previous approaches are to optimize the maximum link utilization and to improve the flow routing in the network. Link selection algorithms are used to optimize distributed estimation and increase network performance. DRL algorithms are used for traffic control and channel rerouting in the network. RL framework helps to improve dynamically routing of flows in the network. So, to improve the efficiency of networks and to increase the overall performance of the SDN, it is needed to use traffic engineering schemes and reinforcement learning methods in the model.

Table 1 An overview on various traffic engineering techniques and routing algorithms

Research work	Addressed issue	Compared with	Result
Xu et al. [1]	Traffic engineering	Shortest path (SP), load balance, DDPG	DRL-TE consistently outperforms DDPG
Wu et al. [2]	Deep reinforcement learning	Convolution neural network (CNN), deep Q learning networking (DQN)	Existing methods are outperformed by the proposed algorithm
Sun et al. [3]	Optimal traffic scheduling	TIDE, DRL-TE	ScaleDRL has better control performance than other DRL solutions
Zhang et al. [5]	Traffic engineering	ECMP	CFR-RL, able to derive to unknown traffic matrices, according to the evaluation findings
Guo et al. [6]	Traffic engineering	OSPF, WA-SRTE, MCF	The proposed ROAR method achieves near-optimal network performance in the hybrid SDN
Chen et al. [7]	Deep reinforcement learning	LL, open shortest path first (OSPF), RL-Routing	As a result of RL-Routing, a host can transfer the large data effectively than OSPF and receive higher rewards as a result
Sun et al. [8]	Routing optimization	Quality of service (QoS)	The effectiveness of TIDE is validated
Rischke et al. [9]	Reinforcement learning	Deep reinforcement learning-traffic engineering (DRL-TE)	A traditional tabular RL technique for the link routing in SDN was developed and assessed
Sun et al. [10]	Deep reinforcement learning	Pinning control	Validate the effectiveness of ScaleDRL

5 Conclusion

The state-of-the-art TE techniques and routing algorithms in SDN and hybrid SDN were analyzed in-depth. For each of the article, the issues addressed, mathematical model or algorithm used along with its core classification is tabulated very clearly for the researchers to have an idea on the literature. The study gives a very elaborated insight.

Table 2 Mathematical model/algorithm used

Mathematical model/algorithm	Purpose of usage	Classification
DRL-TE	The DRL-TE system is greatly reducing end-to-end delay as well as improving total utility	Reinforcement learning algorithms
TCCA-MADDPG algorithms	For traffic control and channel rerouting, the objection function must be optimized	Deep RL method
Link selection algorithm	To optimize the distributed estimation and improve the performance of network by changing topology	Sorting algorithms
Heuristic algorithm	Faster and more efficient approach to solving a problem	Reinforcement learning in SDN
Reinforcement learning algorithm	In computing, a method of determining what actions will be taken by software agents in a given circumstance	Reinforcement learning
Reinforcement learning routing algorithms	RL framework helps improve adaptive routing algorithms	Reinforcement learning
1. Deep deterministic policy gradient algorithms 2. Markov decisions process	The Q-function and the policy are simultaneously learned	Graph theory
Flow routing algorithm	To address the routing flows	Flows routing in SDN
DRL algorithm	Flows weights for selected flows are dynamically adjusted with the DRL algorithm	Pinning control

References

1. Xu Z, Tang J, Meng J, Zhang W, Wang Y, Liu CH, Yang D (2018) Experience-driven networking: a deep reinforcement learning based approach. In: IEEE INFOCOM 2018-IEEE conference on computer communications. IEEE, pp 1871–1879
2. Wu T, Zhou P, Wang B, Li A, Tang X, Xu Z, Ding X (2020) Joint traffic control and multi-channel reassignment for core backbone network in SDN-IoT: a multi-agent deep reinforcement learning approach. IEEE Trans Netw Sci Eng 8(1):231–245
3. Sun P, Lan J, Li J, Zhang J, Hu Y, Guo Z (2020) A scalable deep reinforcement learning approach for traffic engineering based on link control. IEEE Commun Lett 25(1):171–175
4. https://www.fiber-optic-cable-sale.com/will-sdn-change-future-network.html
5. Zhang J, Ye M, Guo Z, Yen CY, Chao HJ (2020) CFR-RL: Traffic engineering with reinforcement learning in SDN. IEEE J Sel Areas Commun 38(10):2249–2259
6. Guo Y, Wang W, Zhang H, Guo W, Wang Z, Tian Y, …, Wu J (2021) Traffic engineering in hybrid software defined network via reinforcement learning. J Netw Comp Appl 103116
7. Chen YR, Rezapour A, Tzeng WG, Tsai SC (2020) RL-routing: an SDN routing algorithm based on deep reinforcement learning. IEEE Trans Netw Sci Eng 7(4):3185–3199
8. Sun P, Hu Y, Lan J, Tian L, Chen M (2019) TIDE: time-relevant deep reinforcement learning for routing optimization. Futur Gener Comput Syst 99:401–409

9. Rischke J, Sossalla P, Salah H, Fitzek FH, Reisslein M (2020) Qr-sdn: towards reinforcement learning states, actions, and rewards for direct flow routing in software-defined networks. IEEE Access 8:174773–174791
10. Sun P, Guo Z, Lan J, Li J, Hu Y, Baker T (2021) ScaleDRL: a scalable deep reinforcement learning approach for traffic engineering in SDN with pinning control. Comput Netw 190:107891
11. Fortz B, Thorup M (2000) Internet traffic engineering by optimizing OSPF weights. In: Proceedings IEEE INFOCOM 2000 conference on computer communications. Nineteenth annual joint conference of the IEEE computer and communications societies (Cat. No. 00CH37064), vol 2. IEEE, pp 519–528
12. Tian Y et al (2020) Traffic engineering in partially deployed segment routing over IPv6 network with deep reinforcement learning. IEEE/ACM Trans Network 28(4):1573–1586. https://doi.org/10.1109/TNET.2020.2987866
13. Guo Y, Wang Z, Yin X, Shi X, Wu J (2014) Traffic engineering in SDN/OSPF hybrid network. In: 2014 IEEE 22nd international conference on network protocols, pp 563–568. https://doi.org/10.1109/ICNP.2014.90
14. Zhang J, Xi K, Luo M, Chao HJ (2014) Dynamic hybrid routing: achieve load balancing for changing traffic demands. In: IEEE international symposium on quality of service'14, pp 105–110
15. Villamizar C (1999) Ospf optimized multipath (ospf-omp). IETF Internet-Draft, draft-ietf-ospf-omp-03.txt
16. Williams RJ (1992) Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach Learn 8(3):229–256
17. Mao H, Alizadeh M, Menache I, Kandula S. Resource management with deep reinforcement learning. In: ACM workshop on hot topics in networks'16, pp 50–56
18. Xu Z, Tang J, Meng J, Zhang W, Wang Y, Liu CH, Yang D (2018) Experience driven networking: a deep reinforcement learning based approach. In: IEEE INFOCOM 2018-IEEE conference on computer communications. IEEE, pp 1871–1879
19. Liu Y-Y, Slotine J-J, Barabási A-L (2011) Controllability of complex networks. Nature 473(7346):167
20. Wu Y, Mansimov E, Grosse RB, Liao S, Ba J (2017) Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In: Advances in neural information processing systems, pp 5279–5288
21. Lin S-C, Akyildiz IF et al (2016) Qos-aware adaptive routing in multi-layer hierarchical software defined networks: a reinforcement learning approach. In: 2016 IEEE international conference on services computing (SCC). IEEE, pp 25–33.
22. Xu Z, Tang J et al (2018) Experience-driven networking: a deep reinforcement learning based approach. In: IEEE INFOCOM 2018-IEEE conference on computer communications. IEEE, pp 1871
23. Wu Y, Mansimov E, Grosse RB, Liao S, Ba J (2017) Scalable trustregion method for deep reinforcement learning using kronecker-factored approximation. In: Advances in neural information processing systems, pp 5279–5288
24. Boyan JA, Littman ML (1993) Packet routing in dynamically changing networks: a reinforcement learning approach. In: Proceedings of 6th international conference on neural information processing systems, NIPS'93. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 671–678. [Online]. Available: http://dl.acm.org/citation.cfm?id=2987189.298727
25. Openflow switch specification version 1.3.5. open networking foundation. [Online]. Available: https://www.opennetworking.org/wpcontent/uploads/2014/10/openflow-switch-v1.3
26. Karakus M, Durresi A (2017) Quality of service (QoS) in software defined networking (SDN): a survey. J Netw Comput Appl 80:200–218
27. Lillicrap TP, Hunt JJ, Pritzel A et al (2015) Continuous control with deep reinforcement learning. Computer Science 8(6):A187
28. Silver D, Lever G, Heess N et al (2014) Deterministic policy gradient algorithms. In: International conference on machine learning, pp 387–395. JMLR.org

29. Tarjan R (1972) Depth-first search and linear graph algorithms. SIAM J Comput 1(2):146–160
30. Guck JW, Van Bemten A, Reisslein M, Kellerer W (2018) Unicast QoS routing algorithms for SDN: a comprehensive survey and performance evaluation. IEEE CommunSurveys Tuts 20(1):388–415
31. Chen JIZ, Smys S (2020) Optimized dynamic routing in multimedia vehicular networks. J Inf Technol 2(3)
32. Bhalaji N (2021) Cluster formation using fuzzy logic in wireless sensor networks. IRO J Sustain Wirel Syst 3(1)
33. Ram GM, Ilavarsan E (2021) Review on energy-efficient routing protocols in WSN. In: Computer networks, big data and IoT