computers

MDPI

Article

Enhancing Reliability in Rural Networks Using a Software-Defined Wide Area Network

Luca Borgianni [1,†], Davide Adami [2,†], Stefano Giordano [1,†] and Michele Pagano [1,*,†]

[1] Department of Information Engineering, University of Pisa, Via G. Caruso 16, 56122 Pisa, Italy; luca.borgianni@phd.unipi.it (L.B.); stefano.giordano@unipi.it (S.G.)
[2] CNIT—Department of Information Engineering, University of Pisa, Via G. Caruso 16, 56122 Pisa, Italy; davide.adami@cnit.it
* Correspondence: michele.pagano@unipi.it
† These authors contributed equally to this work.

Abstract: Due to limited infrastructure and remote locations, rural areas often need help providing reliable and high-quality network connectivity. We propose an innovative approach that leverages Software-Defined Wide Area Network (SD-WAN) architecture to enhance reliability in such challenging rural scenarios. Our study focuses on cases in which network resources are limited to network solutions such as Long-Term Evolution (LTE) and a Low-Earth-Orbit satellite connection. The SD-WAN implementation compares three tunnel selection algorithms that leverage real-time network performance monitoring: Deterministic, Random, and Deep Q-learning. The results offer valuable insights into the practical implementation of SD-WAN for rural connectivity scenarios, showing its potential to bridge the digital divide in underserved areas.

Keywords: SD-WAN; reinforcementlearning; network reliability; smart agriculture; rural connectivity

Citation: Borgianni, L.; Adami, D.; Giordano, S.; Pagano, M. Enhancing Reliability in Rural Networks Using a Software-Defined Wide Area Network. *Computers* **2024**, *13*, 113. https://doi.org/10.3390/computers13050113

Academic Editor: Kannan Govindarajan

Received: 5 March 2024
Revised: 23 April 2024
Accepted: 25 April 2024
Published: 28 April 2024

1. Introduction

Rural Networks enable the flow of information between devices, sensors, actuators, and data centers, allowing farmers to monitor their agricultural operations remotely. This connectivity is essential for deploying various smart agricultural technologies (precision farming, sensor monitoring, and data analytics) that leverage the power of the Internet of Things (IoT) and artificial intelligence (AI). The success of Smart Agriculture Services is related to an adequate design of the Rural Network Connectivity, ensuring that even remote and traditionally unserved agricultural areas are covered. In this scenario, Software-Defined Wide Area Networking (SD-WAN) emerges as a key player in optimizing network performance, ensuring a secure, scalable, and flexible infrastructure for seamless communication across vast agricultural landscapes.

In the past, connecting enterprise sites over long distances relied on dedicated leased lines provided by network operators. However, these lines were associated with high costs and limited speeds, prompting the exploration of alternative technologies for the creation of inter-site connections over public Wide Area Networks (WANs) [1]. Based on packet-switching technology, public WANs matured in the early 1980s and saw continuous development. Technologies like Asynchronous Transfer Mode (ATM) [2], Frame Relay (FR) [3], and Multi-protocol Label Switching (MPLS) [4] were gradually adopted to establish overlaid tunnels for interconnecting enterprise networks (ENs) across sites. MPLS, in particular, has gained popularity due to its ability to ensure QoS through Service Level Agreements (SLAs), which is achieved by establishing dedicated paths within the IP (Internet Protocol) network, leveraging DiffServ (Differentiated Services) technology. However, MPLS has drawbacks such as high bandwidth costs, configuration complexity, and the time required to scale or upgrade paths dynamically.

SD-WAN has emerged as a novel solution in the global enterprise networking landscape, with the aim of offering a low-cost solution that is at least close to the performance of MPLS-based WANs. It leverages the advantages of Software-Defined Networking (SDN) to enhance WANs by separating control and data planes. This approach enables the programming of network devices from a centralized controller, as introduced in [5]. SDN centralizes control plane functions, extracting control logic from underlying routers and switches. This approach simplifies network management and enables innovation through network programmability. While SDN was initially designed to meet the demands of modern computing in data center networks, SD-WAN applies SDN technology to facilitate end-to-end connections between users and networks across the WAN. SD-WAN effectively combines the low cost of Internet access and guarantees a level of availability by introducing a centralized software controller. The SD-WAN overlay architecture is more adaptable to dynamic configurations in response to network conditions than MPLS. This adaptability is facilitated by a controller connected exclusively to edge devices, eliminating the need for direct access to internal WAN equipment, such as providers' routers and switches, to operate an SD-WAN system. This feature is particularly relevant in rural scenarios, in which the simplicity and flexibility of SD-WAN can address the challenges posed by limited resources and infrastructure. Its centralized overlay architecture simplifies network management, minimizing operational complexity, which is particularly advantageous in contexts with limited resources. SD-WAN is the leading technology for many of today's WAN architectures because it guarantees secure, agile, and flexible connectivity among the edge nodes [6]. This is particularly relevant in rural areas where cellular coverage is limited and satellites, Wi-Fi mesh, and fixed xDSL may be present. Furthermore, SD-WAN's ability to adapt dynamically to network conditions is precious in rural areas where connectivity may be unstable or subject to variations. Using a centralized software controller enables dynamic configuration of the SD-WAN architecture to quickly respond to changes in network availability. Its implementation can significantly enhance connectivity and network management in rural environments, facilitating digital transformation even in settings with limited resources.

Tunnel selection is pivotal in SD-WANs, influencing latency, bandwidth, and reliability. Traditional methods often need help to dynamically adapt to changing network-connectivity conditions, making them less effective in highly dynamic environments. The tunnel represents an overlay link, with the CPE (Customer Premises Equipment) possessing a number of outputs corresponding to the technology employed. The CPE is envisioned as an experimental Virtual Network Function (VNF) deployable on generic hardware [7]. It facilitates the automated setup of encrypted tunnels, including IPSec tunnels, across diverse network access technologies like satellite connectivity, mesh WiFi, 4G/5G, xDSL, and fiber optics. In our scenario, a farmer has only one outgoing router, which can serve as the CPE. However, the CPE might also be a more sophisticated device, such as a firewall or a tool for implementing SD-WAN functionalities like connection to the provider, Traffic Engineering (TE), security, and monitoring capabilities.

Reinforcement Learning (RL) [8] has emerged as a powerful paradigm for training agents to make intelligent decisions in dynamic and uncertain environments. In this work, we apply RL techniques to the challenging problem of SD-WAN tunnel selection, aiming to improve the network performance and reliability in rural areas. We introduce a custom SD-WAN environment, implemented using the Mininet and OpenFlow [5] frameworks, to simulate network conditions and evaluate the performance of the RL-based link selection algorithm. The proposed algorithm utilizes a Deep Q-Network (DQN) agent to learn an optimal link selection policy through interactions with the SD-WAN environment.

In this paper, we delve into using SD-WAN technology to ensure comprehensive reliability in areas where a single Long-Term Evolution (LTE) connection may fall short of providing continuous connectivity. By integrating a Low Earth Orbit (LEO) satellite tunnel and employing deep RL for tunnel selection, we propose a solution for rural zones.

To the best of our knowledge, this work is the first to present SD-WAN technology within a rural environment. Let us resume the main contribution of the paper. Firstly, we have developed a simulator for SD-WAN, allowing for the exploration of various scenarios involving multiple tunnels and traffic requests. Our focus lies in the rural scenario, where both LTE and LEO tunnels are available for utilization. We also conduct simulations employing three distinct algorithms to address the tunnel selection problem: random selection, deterministic approaches, and RL-based strategies. The simulation confirms that our proposed methodology shows a promising solution for guaranteeing a reliable connection, underscoring our solution's viability in rural settings.

The rest of the paper is organized as follows. Section 2 presents some of the most relevant works about SD-WAN in remote areas, Traffic Engineering in SD-WAN, and Satellite Internet in remote areas. Section 3 gives an overview of Deep-Reinforcement Learning, focusing on the Deep-Q learning algorithm. Section 4 provides the description of the reference scenario, and Section 5 details the three algorithms. Section 6 describes the implementation and analyzes the results of the experiments. Finally, Section 7 concludes the work.

2. Related Work

Many works exploit poor connectivity resources to improve the QoS and reliability of the rural network scenarios, and in Section 2.1, we present the main contributions in this field. One of the crucial parts of SD-WAN is choosing the proper TE approach. In our work, we exploit RL, and in Section 2.2, we present some of the main RL approaches in an SD-WAN scenario. In Section 2.3, we illustrate satellite internet and how it can be helpful in improving QoS and reliability in rural areas, as well as the original contribution of our work. Finally, Table 1 summarizes the contribution of the different analyzed works.

2.1. SD-WAN in Remote Areas

In SD-WAN, various studies have been conducted to harness its potential for enhancing communication and connectivity in different scenarios. Ref. [9] conducted experiments to evaluate system latencies and compare decentralized earthquake early warning (EEW) design with a centralized EEW approach. These experiments included hypothetical earthquakes, and the results from sixty simulations showed that SD-WAN-based hole-punching architecture with a Transmission Control Protocol (TCP) enabled ideal alerting conditions. Additionally, the decentralized EEW system architecture outperformed the centralized one, saving critical seconds. Another study, ref. [10], focuses on applying SD-WAN in an emergency scenario with a lack of power communication resources. The authors emphasized the importance of emergency communication networks for maintaining electrical grid functionality and highlighted the limitations of optical fiber networks during disasters. The authors optimize the technical solution for hybrid networking, which combines optical fiber networks with the public Internet. SD-WAN implementation significantly improved optical fiber utilization efficiency, serving as an emergency link to restore communication services during cable damage. The authors also provide SD-WAN integration with an operator's customized 4G network. In [11], an IoT-based system was proposed to guide passengers and crew to muster stations during emergency evacuations, offering adaptability and scalability. This system utilizes a private LTE/4G cellular sensor network and an open-source IoT platform to ensure passenger safety during evacuations.

The development of distributed cloud environments necessitates advanced network infrastructure to support network automation, virtualization, high-performance data transfer, and secure access across regional boundaries. The challenges and motivations for distributed cloud environments include cost reduction, redundancy, reliability, geo-replication, and network virtualization. Ref. [12] explores the creation of a cloud-centric and logically isolated virtual network environment using SD-WAN. The authors propose a logically isolated and virtually converged network environment based on virtually dedicated network (VDN) technology over SD-WAN. The scheme aims to offer secured isolation of virtual networks with automated resource convergence, including virtual machines, data centers,

and networks. Finally, ref. [13] discusses using technologies like Low-Power Wide Area Networks (LP-WAN) in SD-WAN. The proposed Emergency Communication System (ECS) combines a wireless 4G/LTE base station and LoRa network, demonstrating efficient data transfer and reliability. LPWAN technology, specifically LoRa, is considered suitable for IoT-based ECS.

2.2. Traffic Engineering in SD-WAN

The authors of [14] propose two TE optimization algorithms to minimize traffic disruptions and the cost related to the use of different underlay networks such as LTE, DSL (Digital Subscriber Line), Cable, etc. On one side, they propose a monitoring algorithm that periodically generates end-to-end probes between nodes in the overlay network to infer link failures and performance degradation in the underlay network. Moreover, it can also infer if two overlay paths share the same underlay congested path. Based on that, the TE algorithms leverage this underlay awareness to re-configure the SD-WAN topology and the overlay routing policies automatically to meet traffic demands and resiliency requirements. The TE optimization algorithms are based on a minimum cost network update (Min-Cost) problem. The goal is to minimize disruptions to the existing overlay network traffic. The authors formulate the first Min-Cost problem as an integer linear programming (ILP) problem that minimizes the network update cost (defined as the total cost of all overlay links used to route the current flows) while satisfying the QoS constraint (defined as the number of congested links shared by the traffic flows). Since this problem is demonstrated to be NP-hard, the authors proposed a greedy algorithm that selects the overlay paths with minimum cost, which approximates the optimal solution in polynomial time. By considering an increasing number of network traffic demands, reconfiguration cost and the number of disturbed flows are considered as performance metrics. According to the results, this approach causes fewer flow disruptions than the baseline, in which every source–destination pair uses the direct link.

The authors of [15] address the impact of traffic transit costs on the Operational Expenditures (OpEx) of SD-WAN service providers, focusing on connections between Data Centers (DCs) and Internet Service Providers (ISPs), along with inter-domain links among ISPs. The primary objective is to introduce a mechanism to minimize these link types' transit costs. Data Center Operators (DCOs) and ISPs typically offer multiple links for transit purposes, such as the proposed algorithm choosing the best link that minimizes the transit costs. Multi-homed ISP architecture involves ISPs providing transfer through several inter-domain links, while DCs enhance resilience by establishing numerous connections to one or more ISPs. Transfer costs vary based on the link speed, medium used, operator agreements (peering, transit, separate for uplink/downlink), and tariff structure (volume-based, 95th percentile-based). Ref. [16] explores the application of SDN to improve energy transactions' reliability, flexibility, and security in smart microgrid systems. The work discusses challenges associated with traditional IP technology, which needs to be improved for real-time business traffic on the Internet. MPLS is suggested as a solution, but it has limitations, particularly in adapting to dynamic requirements. This work introduces SDN as a technology that can offer a dynamic and programmatically efficient network configuration for transactive energy in smart microgrid systems. SDN centralizes network management by abstracting the control plane from the data-forwarding function in networking devices. The paper focuses on an optimized SDN architecture to enhance transactive energy systems' reliability, flexibility, and security within smart microgrids. It proposes the development of such an architecture and evaluates its performance. Ref. [17] proposes an adaptive routing approach that considers the historical performance of network links. As network links often experience failures, effectively managing real-time packet loss between endpoints is crucial for upholding QoS. To address this, the authors enhance a shortest-path algorithm by considering network and reliability parameters, thereby creating intelligent routing within the SD-WAN. This intelligent routing algorithm recommends optimal paths for communication based on the desired latency and reliability. Experimental results show

that their approach successfully demonstrates resilience and efficiency by applying the programmability of SDN for WAN.

In recent research, novel approaches to optimizing network performance and service availability have been proposed by several authors. These studies leverage RL techniques to make intelligent path selection decisions that minimize end-to-end network delay. The work in [18] primarily centers on enhancing SD-WAN performance, with a strong emphasis on improving service availability. The proposed methodology involves multiple stages. Initially, the authors evaluate the performance of baseline threshold-based path selection algorithms. Subsequently, they introduce and implement deep-RL algorithms to address the limitations identified in the baseline algorithms. Specifically, three types of deep-RL algorithms, namely policy gradient, TD-λ, and deep Q-learning, are utilized to predict path conditions regarding end-to-end delay. This predictive capability enables the algorithm to dynamically adapt path selection based on future network conditions. Ref. [19] introduces an innovative approach that leverages variations in Traffic Matrices (TM) as input to train a deep RL algorithm, specifically Soft Actor–Critic. The primary goal is to efficiently extract and encode real-time distributional TM information, especially during dynamic network traffic fluctuations. Advanced deep neural autoencoders are used to extract and encode distributional TM information autonomously at the network's edge. These encoded vectors collectively form a comprehensive Traffic Encoding Matrix (TEM), providing insights into the traffic distribution for each network flow. Subsequently, a path selection algorithm based on Soft Actor–Critic is developed for scalability and suitability for continuous state spaces. In a related context, ref. [20] introduces an innovative SD-WAN framework to facilitate dynamic traffic management within a multi-site enterprise WAN. Real-time network statistics monitoring and a Q-learning-based path selection algorithm are employed to minimize end-to-end service delays and enhance the overall service uptime. The framework is constructed on the Ryu controller and utilizes Mininet to emulate the data plane network. Furthermore, ref. [21,22] have explored the application of deep Q-learning in scenarios involving two and four overlay links with a peering node. Controllers can adapt tunnel selection strategies using deep RL based on real-time network feedback. The outcomes of these studies demonstrate that RL, especially in scenarios with four tunnels, leads to improvements in end-to-end delay and overall QoS.

2.3. Satellite Internet in Remote Area

The study [23] suggested using a network based on LEO satellites to support active network management. Evaluating various IP data services over LEO networks in regular and emergency scenarios revealed that an LEO-based network could meet the new network management solution's bandwidth, availability, and latency requirements. This is because a group of satellites circling in LEO could be utilized to manage and automate a smart grid, meeting the strict time constraints required. It is essential to consider that a geostationary earth orbit (GEO) satellite link has a Round Trip Time (RTT) of 500 ms [24], which may not satisfy the new requirements. Combining the potential of the SD-WAN architecture with the new features that the 6G Satellite promises led to the design of a new architecture, which will be illustrated below.

Ref. [25] explores the integration of SD-WAN and 6G Satellite technology. This integration offers a novel architectural solution to enhance the network performance and reliability of various applications, particularly in remote areas or high-latency environments. The proposed architecture combines the speed and coverage of 6G satellites with the control and flexibility of SD-WAN. It consists of critical components such as 6G Satellites, SD-WAN Edge Devices, SD-WAN Orchestrator, SD-WAN Controller, Security Gateway, Cloud Data Centers, and mobile/IoT devices.

In conclusion, ref. [26] conducts an examination using satellite communication systems to support the IoT. The authors refer to the IoT paradigm as collecting data from sensors or RFID (Radio Frequency ID), and control messages are dispatched to actuators. In numerous application scenarios, sensors and actuators are distributed across extensive geographical

and remote areas where terrestrial access networks are absent. Consequently, satellite communication systems have become paramount for the Internet of Remote Things (IoRT).

Table 1. Summary of related work.

Work	Description	Results
[9]	Improved earthquake early warning using SD-WAN-based hole-punching architecture.	Decentralized EEW system with SD-WAN outperformed the centralized system, saving critical seconds.
[10]	Enhanced emergency communication networks with SD-WAN, combining optical fiber and public Internet.	Reliability and efficiency improved during disasters.
[11]	IoT-based system for guiding passengers and crew to muster stations during emergency evacuations using a private LTE/4G cellular sensor network.	Offers adaptability and scalability for passenger safety during evacuations.
[12]	Creation of a cloud-centric and logically isolated virtual network environment using SD-WAN.	Proposed VDN technology over SD-WAN for secure and efficient network automation.
[13]	Utilizing LPWAN in SD-WAN for efficient data transfer ECS.	Demonstrated efficient data transfer and reliability using LPWAN technology, specifically LoRa.
[14]	Two traffic engineering optimization algorithms to minimize traffic disruptions and costs in SD-WAN.	Proposed Min-Cost problem solution for SD-WAN network updates, minimizing disruptions and reconfiguration costs.
[15]	Mechanism to minimize transit costs in SD-WAN connections between DCs and ISPs.	Algorithm selects the best tunnel to minimize transit costs, considering various link attributes.
[16]	Application of SDN to improve energy transactions' reliability, flexibility, and security in smart microgrid systems.	Proposed optimized SDN architecture enhances reliability and security in energy systems.
[17]	Adaptive routing approach that considers the historical performance of network links in SD-WAN.	Enhancing network reliability and efficiency by considering network and reliability parameters.
[18]	Enhance SD-WAN performance using deep-RL algorithms.	Utilized policy gradient, TD-λ, and deep Q-learning algorithms to predict path conditions regarding end-to-end delay.
[19]	Leveraging variations in TM as input to train a deep RL algorithm for path selection.	Advanced deep neural autoencoders extract and encode distributional TM information.
[20]	Innovative SD-WAN framework for dynamic traffic management in a multi-site enterprise WAN.	Real-time network statistics monitoring and Q-learning-based path selection algorithm for minimizing service delays.
[21,22]	Application of deep Q-learning in scenarios involving two overlay links with a peering node.	Improved end-to-end delay and overall QoS.
[23]	Use of LEO satellites for network management in smart grids.	LEO-based network meets strict time constraints for smart grid automation.
[25]	Integration of SD-WAN and 6G Satellite technology for enhanced network performance and reliability.	Combines the speed and coverage of 6G satellites with the control and flexibility of SD-WAN for various applications.
[26]	Utilizing satellite communication systems for supporting IoT in remote areas.	Satellite communication systems are crucial for IoT in extensive geographical and remote areas.

To the best of our knowledge, this work is the first to use the SD-WAN technology in a rural scenario. In particular, the contributions of this paper are:

- We implement a simulator for SD-WAN with the ability to implement different scenarios with multiple tunnels and requests of traffic.
- We focus on the rural scenario, in which an LTE tunnel and a LEO tunnel are available.
- We conduct simulations using three designed algorithms for the tunnel selection problem: random, deterministic, and RL-based.

3. Reinforcement Learning in SD-WAN

RL [8] is a machine learning paradigm in which an agent learns to make decisions by interacting with an environment in order to maximize a cumulative reward. RL aims to identify optimal actions within a specified environment. This process entails the environment assigning positive or negative rewards based on the actions taken by the agent. The agent plays a key role in decision-making and interaction with the environment, learning from its experiences through iterative interactions with the environment.

At each time step t, the scenario unfolds as follows:

- The environment is characterized by its state s_t.
- The agent executes the action a_t.
- The environment transitions to the state s_{t+1} and conveys the reward r_t to the agent.

These observations, actions, and rewards collectively form the history, defined as $h_t := a_1, s_1, r_1, ..., a_t, s_t, r_t$. This cyclic process represents the core of RL, where the agent continuously refines its decision-making based on the feedback received from the environment. RL has also found applications in trading and finance, healthcare, and gaming, such as in the development of AlphaGo Zero [27], which learned the game of Go from scratch using RL.

Unlike supervised learning, which uses labeled training data, or unsupervised learning, which discovers patterns without guidance, RL requires the agent to actively gather experience about system states, actions, transitions, and rewards to take action in order to maximize a cumulative reward function. In the context of SD-WAN, RL can offer significant advantages in managing and optimizing network performance in dynamic environments. The dynamic nature of network traffic and changing conditions make SD-WAN an ideal candidate for reinforcement learning applications. Figure 1 presents a scheme of RL for the tunnel selection problem in SD-WAN that will be analyzed in the following sections.

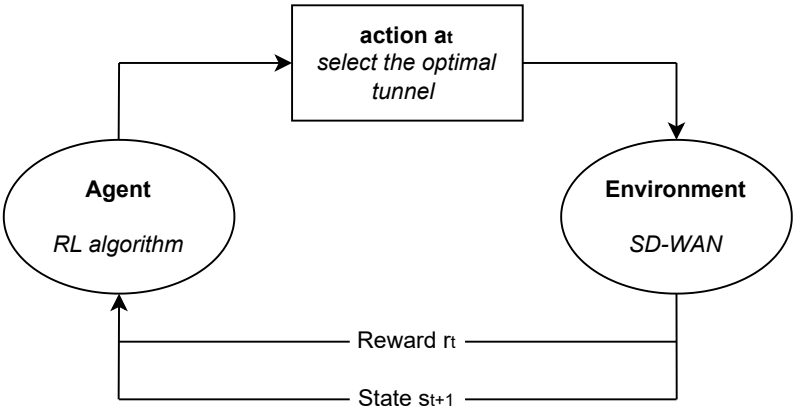


Figure 1. Reinforcement learning mechanism for the SD-WAN scenario

There are several RL algorithms, but analyzing the comparison carried out in [18], we focus on Deep Q-learning because of computational time and performance.

Deep Q-Learning (DQN)

The agent's objective is to perform the propel actions, maximizing future rewards. We adopt the common assumption that future rewards are discounted by a factor γ per time step. The future discounted return at time t is defined as $R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$, where T is the time-step when the simulation terminates. The optimal action-value function $Q^*(s,a)$ in (1) represents the maximum expected return that can be achieved by following any strategy after observing a sequence s and taking action a:

$$Q^*(s,a) = \max_{\pi} \mathbb{E}[R_t | s_t = s, a_t = a, \pi], \tag{1}$$

where π is a policy that maps sequences to actions.

The optimal action-value function in (2) is based on the Bellman equation, which is grounded in the intuition that if the optimal values $Q^*(s_0, a_0)$ for the sequence s_0 at the next time-step were known for all possible actions a_0, then the optimal strategy would be to select the action a_0 that will maximize the expected value of $r + \gamma Q^*(s_0, a_0)$ in the environment E:

$$Q^*(s,a) = \mathbb{E}_{s_0 \sim E}[r + \gamma \max_{a_0} Q^*(s_0, a_0) | s, a]. \tag{2}$$

Many RL algorithms adhere to the basic idea of estimating the action-value function using the Bellman equation as an iterative update: $Q_{i+1}(s,a) = \mathbb{E}[r + \gamma \max_{a_0} Q_i(s_0, a_0) | s, a]$. However, this approach becomes impractical in practice, as the action-value function is estimated separately for each sequence without generalization. Instead, a function approximator, typically a Q-network, is commonly employed to estimate the action-value function $Q(s,a;\theta) \approx Q^*(s,a)$, where θ represents the network weights [28].

Training a Q-network involves minimizing a sequence of loss functions $L_i(\theta_i) = \mathbb{E}_{s,a \sim \rho(\cdot)}[(y_i - Q(s,a;\theta_i))^2]$, where $y_i = \mathbb{E}_{s_0 \sim E}[r + \gamma \max_{a_0} Q(s_0, a_0; \theta_{i-1}) | s, a]$ is the target for iteration i, and $\rho(s,a)$ represents a probability distribution over sequences s and actions a referred to as the behavior distribution. The parameters from the previous iteration θ_{i-1} are fixed when optimizing the loss function $L_i(\theta_i)$. Note that the targets depend on the network weights; this contrasts with the targets used for supervised learning, which are fixed before learning begins.

Differentiating the loss function concerning the weights, we arrive at the gradient in (3):

$$\nabla_{\theta_i} L_i(\theta_i) = \mathbb{E}_{s,a \sim \rho(\cdot); s_0 \sim E}\left[\left(r + \gamma \max_{a_0} Q(s_0, a_0; \theta_{i-1}) - Q(s,a;\theta_i)\right) \nabla_{\theta_i} Q(s,a;\theta_i)\right]. \tag{3}$$

Rather than computing the full expectations in the above gradient, it is often computationally reasonable to optimize the loss function by stochastic gradient descent. If the weights are updated after every time step, and expectations are replaced by single samples from the behavior distribution ρ and the environment E, respectively, then we arrive at the familiar Q-learning algorithm [29].

It is crucial to note that this algorithm is model-free, meaning it directly solves the RL task using samples from the environment E without explicitly constructing an estimate of E. Furthermore, it is off-policy, as it learns about the greedy strategy $a = \max_a Q(s,a;\theta)$ while following a behavior distribution that ensures sufficient state space exploration. In practical applications, the behavior distribution is often selected using an ϵ-greedy strategy, where the agent follows the greedy strategy with a high probability of $1 - \epsilon$ and selects a random action with a probability of ϵ.

4. Reference Architecture

We introduce the reference scenario (see Figure 2), providing a high-level, generalized system description. The system operates within a rural environment, with N peripherals connected with K tunnels to a central site. Each peripheral could connect M different users with different types of traffic requests. This architecture aims to enhance efficient data

exchange and reliable communication in rural landscapes where conventional infrastructure might be limited. The system leverages SDN as an enabler in the SD-WAN framework. SDN offers a centralized control mechanism, enhancing the management and orchestration of network resources. The central controller is the focal point for decision-making, allowing for dynamic adjustments in response to varying network conditions. A crucial element in this system is the utilization of Tunnel Selection Algorithms (TSAs). These algorithms are pivotal in efficiently forwarding traffic through the network tunnels. Their primary function is to assess and select the most optimal tunnel based on specific criteria, bandwidth availability, overall network load, tunnel availability, and traffic type. In this work, we will consider three different algorithms, and the description of each algorithm will be presented in Section 5.

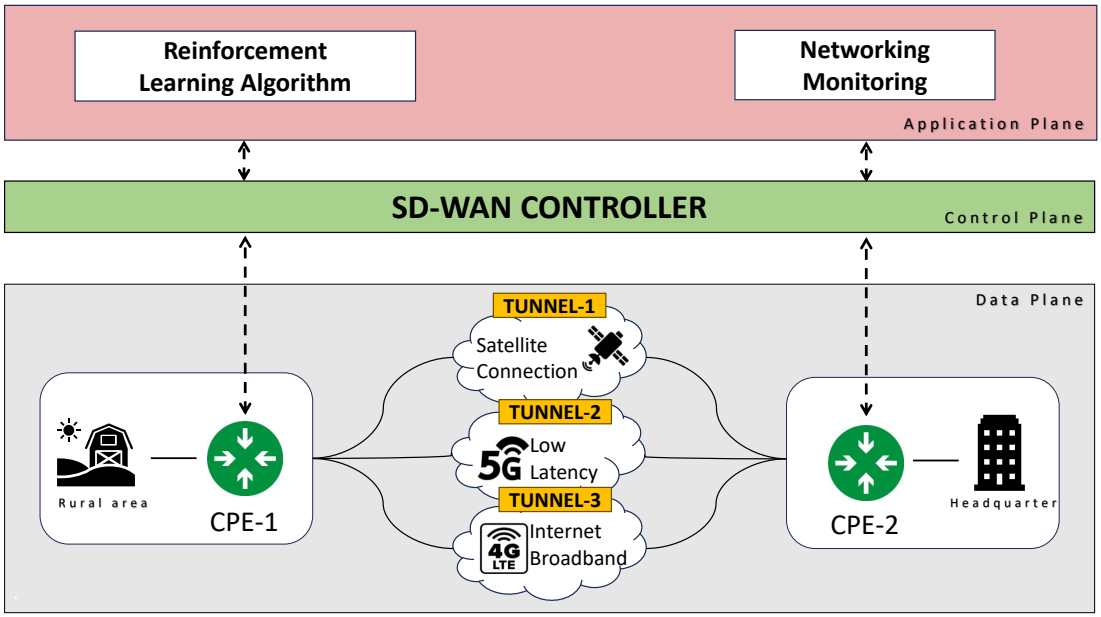


Figure 2. An example of SD-WAN in a rural scenario that exploits three tunnels from different technologies.

To contextualize, let us present a specific scenario (see Figure 3) where we considered four rural sites ($N = 4$) connected to a central data center; this is based on the general SD-WAN architecture. Each rural area is connected through an SD-WAN CPE and two tunnels ($K = 2$), and may generate two types of traffic requests ($M = 2$) that we define as type A and type B. The first tunnel is an LTE tunnel, while the other represents a LEO tunnel. It is worth noting that when we talk about SD-WAN, the management will be at the end-to-end level, and the underlay routing within each tunnel will not be controllable.

The LTE tunnel and LEO tunnel diverge significantly in terms of reliability and QoS. LTE offers a terrestrial wireless communication standard known for its reliability. However, its performance may vary, especially with intermittent connectivity and fluctuating QoS. On the other hand, the LEO tunnel provides constant connectivity, ensuring a stable connection, but it still exhibits variable QoS based on factors such as weather conditions, equipment quality, the satellite technology used, and the geographical location. While generally reliable, the LTE tunnel experiences intermittent connectivity in the rural scenario we considered. This means that the connection may not be consistently available, introducing variability in its QoS, making it crucial to assess these factors when considering the suitability of this tunnel for specific types of traffic. In contrast, the LEO tunnel offers constant connectivity, ensuring a reliable and stable connection. However, its QoS remains variable, influenced by the factors listed above. The choice between LTE and LEO tunnels depends on the nature of the traffic and the specific QoS requirements.

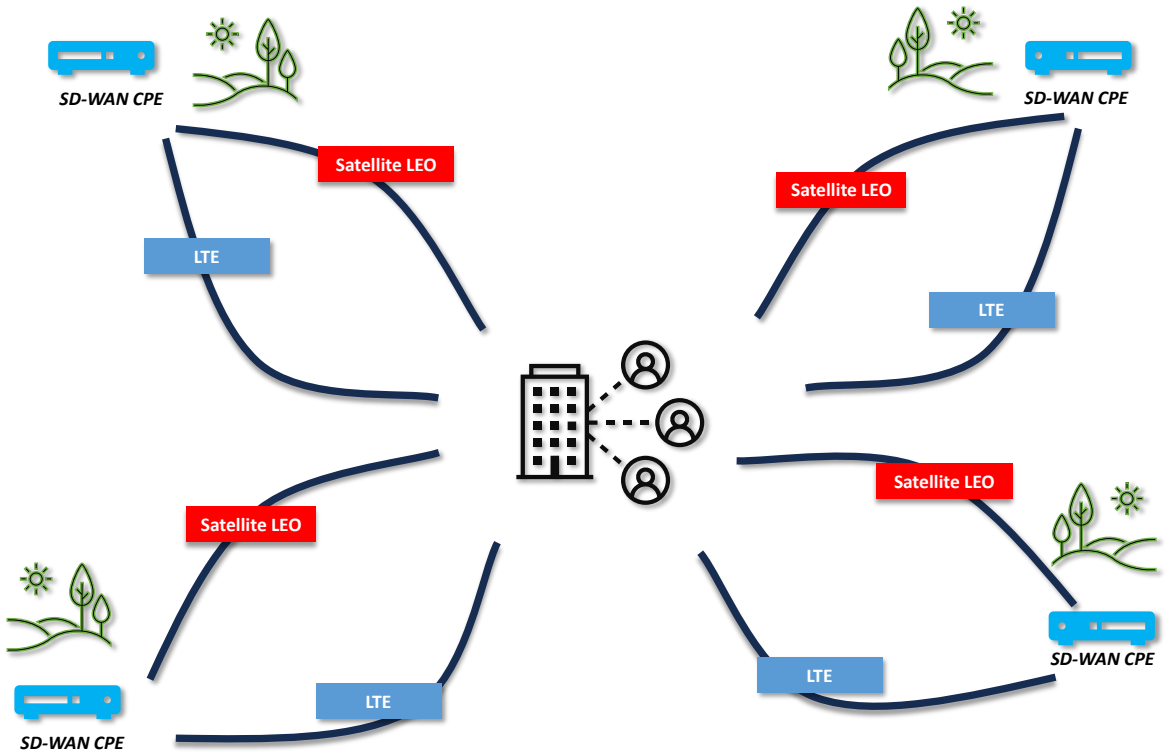


Figure 3. Rural SD-WAN environment considered with LTE and LEO tunnels.

5. Tunnel Selection Algorithms

We present three different algorithms to evaluate the efficacy of the proposed solution with a particular focus on rural areas: random, deterministic, and Deep Q-learning-based. For all algorithms, the selection of the tunnel is exclusive, and no forms of multipath or redundancy are allowed. In our analysis, we examined the collective traffic generated by all CPEs, and the algorithms presented herein are intended to be applied individually to each CPE.

5.1. Random

The first Algorithm 1 adopts a random approach to tunnel selection, reflecting the unpredictability inherent in network conditions. A simple function that can be allocated in the controller randomly selects an action that corresponds to the selection of a tunnel. Traffic will be forwarded over one of the K tunnels with the same probability independently of the tunnels' statues and bandwidth.

Algorithm 1 Random

1: Initialize system parameters: *user_traffic*, *available_bw*, *traffic_type*
2: **for** traffic event **do**
3: Randomly select *action* ▷ Among *K* Tunnels
4: **end for**

5.2. Deterministic

As regards the deterministic algorithm, we introduce specific rules for tunnel selection based on the user traffic and available bandwidth for each tunnel. In particular, we compare the user traffic bandwidth with the available bandwidth in the first tunnel that corresponds to the default tunnel. As shown in the pseudo-code of the Algorithm 2, if the default tunnel

does not have enough bandwidth to satisfy the user traffic request, the algorithm will select the tunnel with the highest available bandwidth.

Algorithm 2 Deterministic with K Tunnels

```
Initialize system parameters: user_traffic, available_bw, traffic_type
for traffic event do
    if user_traffic ≤ available_bw_tunnel_1 then
        action = 1                                          ▷ Default tunnel
    else
        max_bw = 0
        for i = 1 to K do
            if available_bw_tunnel_i > max_bw then
                max_bw = available_bw_tunnel_i
                action = i                                  ▷ Select tunnel i
            end if
        end for
    end if
end for
```

5.3. DQN-Based

The third Algorithm 3 introduces the RL-based approach that leverages DQN with the aim of selecting the propel tunnel in real-time. The most important advantage of using RL is that the algorithm will learn in real-time, as described in Section Deep Q-Learning (DQN). In other words, it is possible to forecast the overcoming of the QoS threshold, and it will change the tunnel not only when the instant SLA threshold is not satisfied but also before that situation occurs. The primary objective of DQN is to learn and adapt the tunnel selection, particularly in scenarios in which the presence or absence of a primary connectivity becomes a critical determinant. When the existing tunnel bandwidth is insufficient to meet traffic requirements, DQN strategically anticipates the impending challenge and proactively recommends a tunnel change. This proactive approach enhances the overall network performance and contributes to resource efficiency.

DQN combines Q-learning (which updates Q-values using the Bellman equation, ensuring the model learns from historical experiences) with deep neural networks to approximate the Q-value function, representing the expected return for a given action in a specific state. The following parameters characterize the neural network:

Input Layer: The input layer is designed to match the environment's state space size, capturing the relevant features required for decision-making. In particular, the implemented input layer has as many input nodes corresponding to the dimension of the state with a dimension of $K + 2$ where K is the number of tunnels (we see this in our simulation scenario $K = 2$, and so the input layer will have four nodes).

Hidden Layers: There are two hidden layers, each comprising 24 neurons and activated by the Rectified Linear Units (ReLU) function. Each neuron in this layer receives inputs from the three features previously listed. The hidden layer transforms the input information into a higher-dimensional space through the weighted connections and activation function, facilitating the neural network's ability to learn and generalize from the observed states.

Output Layer: The output layer of the neural network is designed to reflect the two possible actions that the agent can take. This layer comprises two neurons, aligning with the count of available actions, and serves as the neural network's interface to provide Q-value estimates for the available actions. Furthermore, it uses a linear activation function for the output layer to directly obtain the expected Q-values for each action without any nonlinear transformation.

Loss Function and Optimizer: The Mean Squared Error (MSE) loss function quantifies the difference between predicted and actual Q-values. With a learning rate of 0.001, the Adam optimizer [30] orchestrates the iterative refinement of model parameters during the training process.

Algorithm 3 Deep Q-Learning with experience replay

Parameters: Replay Memory Size, Batch Size, MAX_EPISODES, MAX_STEPS, Exploration Min, Exploration Decay
D = Replay Memory Size
for episode = 1 to MAX_EPISODES **do**
    Reset environment to initial state s
    for step = 1 to MAX_STEPS **do**
        Select action a using ϵ-greedy policy based on current Q-values
        Execute action a in the environment, observe next state s' and get reward r
        Store (s, a, r, s') in replay memory D
        if length$(D) \geq$ Batch Size **then**
            Sample a random batch (s, a, r, s') from D
            for (s, a, r, s') in batch **do**
                Predict Q-values for state s
                Update Q-value for action a in Q-values
                Train the model with state s and updated Q-values
            end for
            $\epsilon \times =$ Exploration Decay ▷ Update exploration rate
            $\epsilon = \max($Exploration Min$, \epsilon)$
        end if
    end for
end for

This RL model uses experience replay, in which experiences are stored and randomly sampled, enhancing their learning efficiency. The Q-network's weights are updated to minimize the disparity between predicted and target Q-values. A target network stabilizes learning, and a mean squared error loss function quantifies the difference between predicted and target Q-values. The DQN implementation incorporates a discount factor (γ) to balance immediate and future rewards, emphasizing long-term gains. The general state s has a dimension of $K + 2$, where K is the number of tunnels. The first element is 0 when the default tunnel is not available; from 1 to K are 0 when the k-th tunnel does not have enough available bandwidth with respect to the user traffic; and the last element represents the traffic type. In Section 6, we will describe in detail the state for a specific simulated scenario. As described previously, the agent's objective is to maximize the cumulative sum of reward-based learning from past experience. In this study, the reward function is designed to let the agent learn about the tunnel's availability and the bandwidth's availability. The chosen parameters play a key role in defining the behavior of the DQN agent (see Table 2).

Table 2. Parameter configuration for the algorithm.

Parameter	Value
Discount factor γ	0.95
Replay memory	1000
Batch size	32
Exploration Rate	starts at 1.0 and decays to 0.01

Discount Factor (γ): The discount factor, set to 0.95, emphasizes future rewards. A high value (close to one) indicates a strategic focus on long-term gains.

Replay Memory Size: With a replay memory size of 1000, the agent leverages past experiences to enhance learning. This memory enables the agent to draw from diverse historical scenarios, mitigating potential biases associated with learning from consecutive experiences and contributing to the stability of the training process.

Batch Size for Training: Training occurs in batches of 32 experiences randomly sampled from the replay memory. This batching strategy prevents overfitting to specific experiences

and promotes smoother learning, enhancing the agent's ability to generalize across different situations.

Exploration Rate: The exploration–exploitation trade-off is managed through an ϵ-greedy strategy [31]. Exploration is gradually reduced over time, transitioning from random exploration to exploitation of learned policies. During the action selection phase, the agent considers two scenarios:

- Exploration (with probability ε): The agent randomly explores new actions instead of following the current policy, generating a random number from a uniform distribution between 0 and 1. If the random number is less than ϵ (the exploration rate), a random action is chosen from all possible actions.
- Exploitation (with probability $1 - \epsilon$): The agent exploits the knowledge gained by selecting the action with the maximum Q-value for the current state.

Starting with an exploration rate of 1.0, the rate gradually decays exponentially over time with a decay value of 0.995, reaching a minimum value of 0.01.

6. Implementation and Simulation Results

In this section, we present the implementation of the SD-WAN architecture in a simulated environment. Moreover, we implement and evaluate the algorithms presented in Section 5 in a specific rural scenario. In particular, we consider a rural site where the single LTE connection is not able to guarantee complete spatial and temporal coverage and provide adequate reliability. An additional LEO satellite tunnel is considered. We considered two types of traffic (Type A and B) generated as in the Section 6.1. In both cases, ensuring reliable data transmission is critical in remote and rural areas. Losing packets can have significant consequences, especially for critical applications like file transfers, web browsing, and emails.

In this results section, we will show that by adding an LEO tunnel with a proper tunnel selection allowed by SD-WAN, we can guarantee continuous coverage in space and time, allowing uninterrupted communication compared to if we had only LTE for our type of traffic requests. We will make a comparison with the scenario in which LTE is the only connection solution, and we will also compare the various algorithms to see which one performs better.

6.1. Simulated Bandwidth Generation

We consider Background Traffic (BT) and User Traffic (UT) in our simulation. Available Bandwidth is derived by subtracting the BT from the total Tunnel Bandwidth, which is constant. We considered BT=0 for the LEO tunnel in our simulated scenario. BT represents the persistent flow of data over the network, which is typically attributed to long-term connections like long-lived TCP sessions originating from peripheral sites or other users who share the tunnel links. This traffic has been generated according to a model inspired by [32]. Figure 4 shows the BT of one simulation.

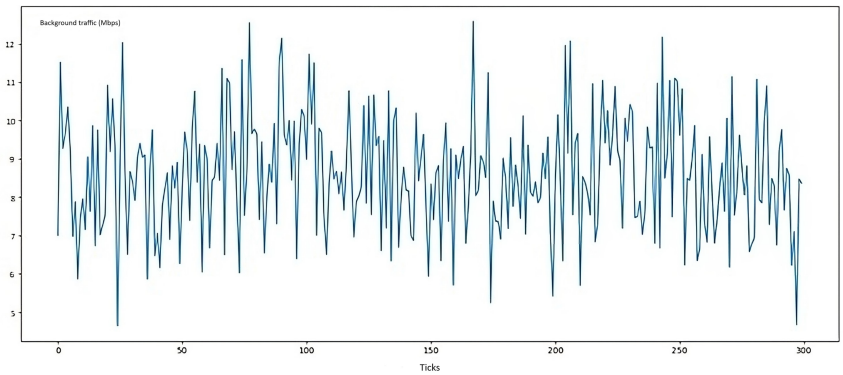


Figure 4. Background traffic of LTE tunnel for one simulation.

UT is a short-term traffic component that takes into account new traffic demands coming from the end-users at the peripheral site: this traffic has been generated according to a truncated normal distribution [33]. For traffic type 'Type A', the request of traffic is generated using a normal distribution with a mean of 15 Mbps (μ_a) and a standard deviation of 2 Mbps (σ_a). On the other hand, for traffic type 'Type B', the generation uses a mean of 5 Mbps (μ_b) and a standard deviation of 2 Mbps (σ_b), reflecting common bandwidth values and variations encountered in real-world SD-WAN environments [34]. We consider 2/3 of the traffic Type A and 1/3 Type B. The truncation is introduced so that negative values are not considered even if the probability is very small. With a mean of 5, we have 6.2×10^{-3} probability of having negative values and 3.2×10^{-14} when the mean is 15.

6.2. Simulation Scheme

The SD-WAN simulator was fully developed in Python, allowing us to integrate some RL libraries, which will be specified later. The implementation allows the creation of a scenario with multiple types of traffic requests and multiple user sites that communicate with a data center with multiple tunnels. In this paper, we considered a specific scenario and three TSAs, but the simulation scenario can be easily changed and adapted to other situations.

According to the scenario previously described and the algorithms in Section 5, we performed some tests to evaluate how the three algorithms behave in the simulated scenario.

We implemented a DQN agent to learn the optimal tunnel selection policy. The agent's neural network model uses the Keras library [35] with a TensorFlow [36] backend. For the RL algorithm, we employed OpenAI Gym [37], a Python library designed for developing and comparing RL algorithms. It provides a standardized API (Application Programming Interface) for communication between learning algorithms and environments and a set of environments that comply with the established API. The random algorithm will select, with a probability of 0.5, one of the two available tunnels. In Figure 5, we present the general scheme of the simulation. Let us summarize the simulation steps:

1. Select the type of traffic (Type A and Type B) and generate BT according to Section 6.1.
2. The three TSAs (Section 5) select the tunnel (LTE or Satellite LEO).
3. Generate the LTE status (present or not). In order to simulate a rural scenario, we considered the LEO tunnel as always being present, while the LTE tunnel is present with a probability of 0.8.
4. We update the QoS statistics, taking into account the selected tunnel and the LTE status.
5. Repeat starting from (1) until the SD-WAN simulation stops.

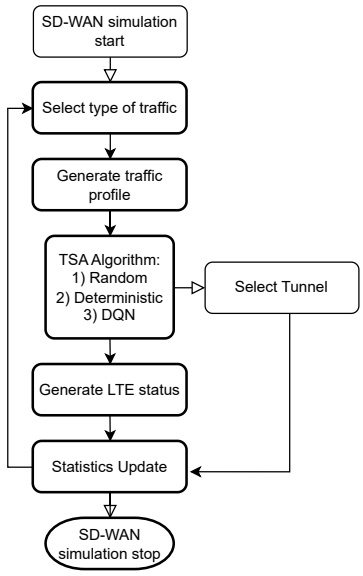
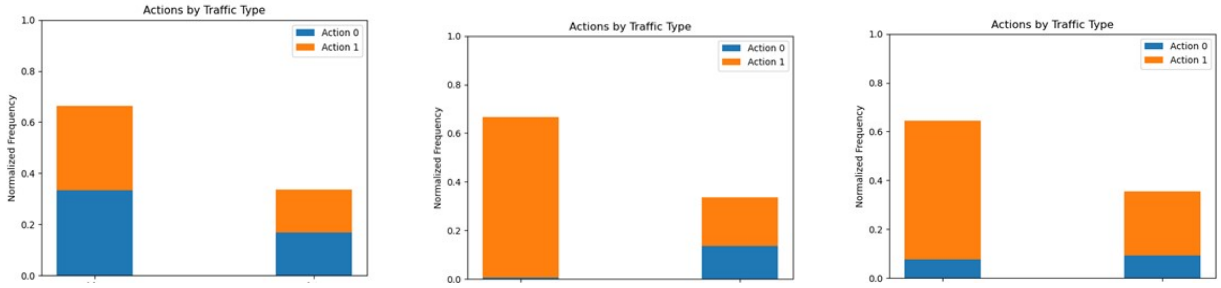


Figure 5. General scheme for the tunnel selection algorithms in the SD-WAN scenario.

In the following sections, we present some results based on the scenario implemented and the three different algorithms.

6.3. Tunnel Selection

Figure 6 illustrates the normalized frequency of occurrences for action 0 (LEO tunnel selection) and action 1 (LTE tunnel selection), considering the two different traffic types and the three algorithms.



(**a**) Actions for random algorithm (**b**) Actions for deterministic algorithm (**c**) Actions for DQN algorithm

Figure 6. Number of actions for each type of traffic in the three algorithms.

Figure 6a confirms that the random algorithm does not consider any criteria as contributing toward best performance. We can see that the deterministic (Figure 6b) and DQN approach (Figure 6c) present different behaviors of the selected tunnel, in accordance with the goal of selecting the best tunnel according to the available bandwidth.

6.4. Reward Function

The design of the reward function is a crucial element of RL. By providing positive or negative values with different absolute values, we present different situations that the algorithm should reward or penalize. The general rule is to force the algorithm to learn the forwarding of the traffic in order to ensure minimal loss, avoiding the LTE tunnel when it is down and considering the bandwidth availability for both tunnels. The difference in the values is justified by performing an action with 100% loss (LTE not present), <100% loss, or 0% loss. For example, negative values are justified by the fact that we penalize the case in which the chosen tunnel does not have enough bandwidth and thus is unnecessarily occupied by taking up bandwidth for another traffic request. In Table 3, we present the instantaneous value for the specific scenario simulated in accordance with the action taken and the state $s = [a, b, c, d]$, where a is 0 when the LTE tunnel is not available, b and c are 0 when the LTE and LEO tunnel, respectively, do have not enough bandwidth available with respect to the user traffic, and d is 0 when the traffic type is Type B. We excluded the inconsistent states $[0, 1, *, *]$ in which the LTE tunnel is not present but has sufficient bandwidth in accordance with the proposed simulation model.

As shown in Table 3, let us explain three examples according to the action selected in a particular state:

- State$[0, 0, 1, *]$ with action 0: In this case, the reward is highly negative (-4) because the algorithm has selected the LTE tunnel that is not present with a consequent loss of 100%.
- State$[1, 0, 0, *]$: In the case of action 0, the reward is slightly negative (-2) because the algorithm has selected the LTE tunnel that does not have enough bandwidth (loss < 100%). But on the other hand, action 1 will also have a loss. In this last case, we assign a positive reward (+2) because, as a general strategy, the algorithm should learn to forward to the LEO tunnel when neither of them has sufficient bandwidth.
- State$[1, 1, 1, 1]$ with action 0: The type of traffic is considered only when both tunnels guarantee sufficient bandwidth. In this case, we force the algorithm to forward type A to the LEO tunnel. In this case, we assign a very high positive because not only is the loss 0%, but the type of traffic has been forwarded to a preferential tunnel.

Table 3. Reward States.

Reward	Action	States
−3	0	$[0,0,0,*]$, $[1,0,1,*]$
−3	1	$[1,1,0,*]$
-4	0	$[0,0,1,*]$
+3	1	$[0,0,0,*]$
+4	1	$[0,0,1,*]$, $[1,0,1,*]$, $[1,1,1,1]$
+4	0	$[1,1,0,*]$, $[1,1,1,0]$
+2	1	$[1,0,0,*]$
−2	0	$[1,0,0,*]$
+8	0	$[1,1,1,1]$
+8	1	$[1,1,1,0]$

6.5. Learning Trend

In order to show the learning trend of the DQN-based algorithm, Figure 7 presents the behavior of the reward for one simulation. Analyzing reward trends offers an understanding of the adaptive learning dynamics of the network. The reward in the DQN approach shows an upward trend over time. This phenomenon is the direct consequence of the inherent nature of DQN, wherein the algorithm learns from its own actions with the goal of maximizing the reward. These results underscore the effectiveness of DQN in dynamic learning in accordance with network conditions, emphasizing the importance of learning-based approaches in our tunnel selection scenario in SD-WAN.

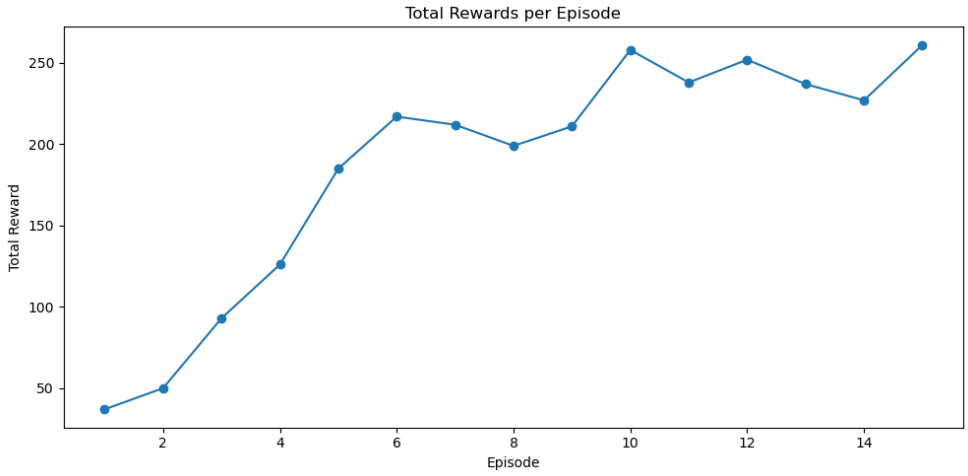


Figure 7. Reward trend deep Q-learning.

6.6. Comparison with LTE-Only Scenario

As previously described, our work aims to showcase that our proposed and implemented SD-WAN architecture can be effective in a rural area where an LTE connection exhibits intermittent coverage. In Table 4, we conducted a comparison between the SD-WAN scenario, where three algorithms (DQN, Deterministic, and Random) dynamically select one tunnel, as outlined in the preceding sections, and a scenario featuring only LTE connectivity. In terms of how this is implemented in our system, SD-WAN inherently guarantees 100% coverage, since at least one tunnel is always available. In contrast, we considered the same scenario while excluding diversion to the LEO tunnel, resulting in a coverage of 72%. Table 4 also introduces the Traffic Excess Factor (TEF) metric, representing the difference (in %) between the UT and the available bandwidth normalized with the

total UT only at the steps with losses (i.e., when the difference is positive). In other words, it serves as a kind of "pseudo-packet loss". In this context, we compared the performance of the three algorithms within the SD-WAN framework. The results confirm that DQN exhibited the best values, leveraging its ability to learn the evolving tunnel availability over time and predict traffic forwarding to tunnels that may soon become bandwidth-insufficient. The lower values of TEF for DQN (8%) and deterministic (10%) show that it is possible to control the tunnel selection in order to avoid the tunnel with limited capabilities. The point that we would like to highlight is that the DQN algorithm is able to learn a policy in order to achieve an improvement in terms of the metric presented, and it can reach or overcome the performance of a ruled-based algorithm as the deterministic one.

Table 4. Comparison with LTE-only scenario.

Metrics	SD-WAN (LTE and LEO)	LTE (Only)
Availability	100%	72%
Traffic Excess Factor (TEF)	8% (DQN) 10% (Deterministic) 16% (Random)	19%

It is important to note that all three algorithms outperformed the LTE-only case, where the TEF metric considers bandwidth unavailability even when the tunnel itself is not accessible.

6.7. Ratio of Steps with Sufficient Bandwidth

We define the "ratio of steps with sufficient bandwidth" as the ratio between the number of steps with the tunnel bandwidth that does not exceed the residual bandwidth and the total number of steps in each episode. In other words, it quantifies the percentage of times wherein the chosen tunnel provides adequate bandwidth. Starting from the random algorithm, we can notice from Figure 8 (orange line) that this ratio has lower values with respect to the other two algorithms, and this behavior reflects its random nature as the tunnel selection is performed without following any criteria and we cannot observe any particular advantage in using a backup tunnel. On the other hand, the deterministic algorithm (blue line) presents higher values, with the lowest peaks always greater than the random. As regards the DQN (gray line), it is noteworthy to observe the system's capability to learn and dynamically opt for the most suitable tunnel based on the previous network conditions. This adaptive behavior highlights the effectiveness of DQN in autonomously adjusting tunnel selection in response to varying network conditions, thereby mitigating instances where the bandwidth requirements exceed the capabilities of the chosen tunnel with a consequence of increasing the trend over time. Both the deterministic and DQN algorithms reach a stable value of over 95% of the metric considered, indicating their effectiveness. In contrast, the random approach shows a peak of 64%, which differs significantly from the DQN and deterministic results.

Finally, Table 5 compares the throughput of the algorithms that we are considering in this work. The throughput achieved with the DQN algorithm has been considered in two specific episodes: the first and the last after training. As a reference, the random algorithm achieves a throughput of 9.3 Mbps, while the deterministic algorithm achieves a value of 13.2 Mbps. In the first episode of DQN, the throughput is 8.7 Mbps, indicating that the initial performance of the algorithm is comparable to a random selection (exploration phase of RL). However, after only 15 episodes of training, the throughput significantly improves to 13.7 Mbps, demonstrating the effectiveness of learning through RL, overtaking both random and deterministic methods.

Table 5. Comparison of the throughput.

DQN (First Episode)	DQN (Last Episode after Training)	Random	Deterministic
8.7 Mbps	13.7 Mbps	9.3 Mbps	13.2 Mbps

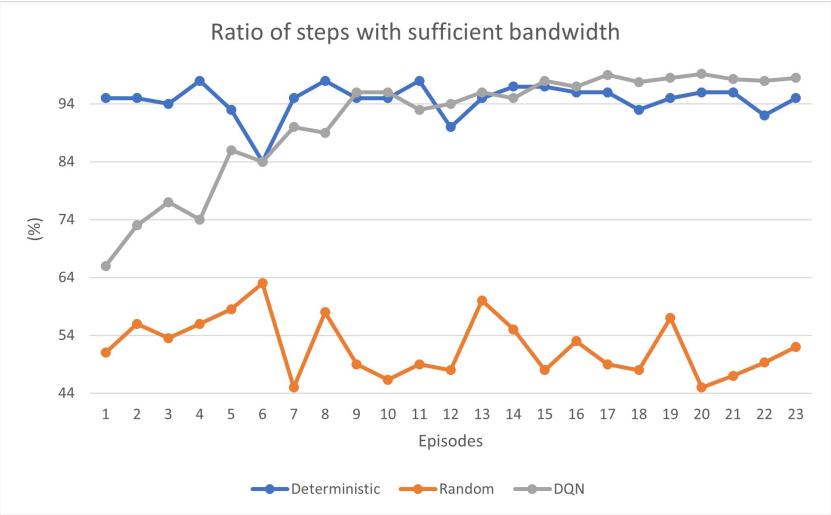


Figure 8. Ratio of steps with sufficient bandwidth.

7. Conclusions

Our paper presents an innovative approach that utilizes SD-WAN architecture to address the challenge of providing reliable network connectivity in rural areas with limited infrastructure. Through our study and the SD-WAN simulator implemented, we examined the efficacy of three tunnel selection algorithms (Deterministic, Random, and DQN) that leverage real-time network performance monitoring in scenarios where resources are constrained to solutions like intermittent LTE and LEO satellite connections. For the DQN algorithms, we also designed a reward function for the scenario considered. The findings show the potential of SD-WAN solutions to mitigate the digital divide in undeserved regions. We show how different algorithms can impact the performance of the SD-WAN system designed for rural and remote environments. In particular, we introduce the TEF metric (less values mean better values) to compare the LTE-only scenario and the SD-WAN(LTE and LEO scenario), achieving an improvement from 19% in the LTE scenario to 8% in the SD-WAN scenario with DQN. Moreover, the throughput finding presents the learning trend of the DQN algorithm and how it overcomes the other solutions after training. Finally, we compare the metric Ratio of steps with sufficient bandwidth to obtain a complete overview of the three algorithms. Hence, we can conclude that, by enhancing reliability and efficiency, SD-WAN emerges as a promising solution to support connectivity in rural and remote areas.

Author Contributions: Conceptualization, L.B., D.A., S.G. and M.P.; methodology, L.B., D.A., S.G. and M.P.; software, L.B., D.A., S.G. and M.P.; validation, L.B., D.A., S.G., and M.P.; formal analysis, L.B., D.A., S.G. and M.P.; investigation, L.B., D.A., S.G. and M.P.; resources, L.B., D.A., S.G. and M.P.; data curation, L.B., D.A., S.G. and M.P.; writing—original draft preparation, L.B. and D.A.; writing—review and editing, L.B., D.A., S.G. and M.P.; visualization, L.B., D.A., S.G. and M.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: The authors would like to thank Setayesh Ghadir and Delaram Ghadir for their work in support of the experimental activities. This work was partially supported by the Italian

Ministry of Research (MUR) in the framework of the CrossLab & Forelab Projects (Departments of Excellence) and by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001-program "RESTART"). Finally, we acknowledge the use of ChatGPT (accessed on 3 February 2024) (https://chat.openai.com/) for English correction and rephrasing in some sections of the manuscript. It was not used to generate new content or ideas; it was employed only as a tool to enhance the smoothness and correctness of the existing text.

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

Abbreviations

The following acronyms (listed in alphabetical order) are used in this manuscript:

AI	Artificial Intelligence.
API	Application Programming Interface.
ATM	Asynchronous Transfer Mode.
BT	Background Traffic.
CPE	Customer Premises Equipment.
DQN	Deep-Q Learning.
DSL	Digital Subscriber Line.
EEW	Earthquake Early Warning.
ECS	Emergency Communication System.
FR	Frame Relay.
GEO	Geostationary Earth Orbit.
ILP	Integer Linear Programming.
IP	Internet Protocol.
ISP	Internet Service Provider.
LEO	Low Earth Orbit.
LP-WAN	Low-Power Wide-Area Networks.
LTE	Long-Term Evolution.
MPLS	Multi-Protocol Label Switching.
MSE	Mean Squared Error.
RL	Reinforcement Learning.
RFID	Radio Frequency ID.
RTT	Round Trip Time.
SD-WAN	Software Defined Wide Area Network.
SDN	Software Defined Networking.
SLA	Service Level Agreements.
TE	Traffic Engineering.
TEM	Traffic Encoding Matrix.
TEF	Traffic Excess Factor.
TCP	Transmission Control Protocol.
TM	Traffic Matrix.
TSA	Tunnel Selection Algorithm.
UT	User Traffic.
VDN	Virtually Dedicated Network.
VNF	Virtual Network Function.
WAN	Wide-Area Network.

References

1. Boger, P. *Connecting Networks Companion Guide*; Pearson Education: London, UK, 2014.
2. Le Boudec, J.Y. The asynchronous transfer mode: A tutorial. *Comput. Netw. Isdn Syst.* **1992**, *24*, 279–309. [CrossRef]
3. Buckwalter, J.T. *Frame Relay: Technology and Practice*; Addison-Wesley Professional: Boston, MA, USA, 2000.
4. Davie, B.S.; Rekhter, Y. *MPLS: Technology and Applications*; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA, 2000.
5. McKeown, N.; Anderson, T.; Balakrishnan, H.; Parulkar, G.; Peterson, L.; Rexford, J.; Shenker, S.; Turner, J. OpenFlow: Enabling innovation in campus networks. *ACM SIGCOMM Comput. Commun. Rev.* **2008**, *38*, 69–74. [CrossRef]
6. Troia, S.; Zorello, L.M.M.; Maier, G. SD-WAN: How the control of the network can be shifted from core to edge. In Proceedings of the 2021 International Conference on Optical Network Design and Modeling (ONDM), Gothenburg, Sweden, 28 June–1 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–3.

7. Woodyatt, J. Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service. Technical Report. 2011. Available online: https://www.rfc-editor.org/rfc/rfc6092 (accessed on 10 October 2023).
8. Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **1996**, *4*, 237–285. [CrossRef]
9. Prasanna, R.; Chandrakumar, C.; Nandana, R.; Holden, C.; Punchihewa, A.; Becker, J.S.; Jeong, S.; Liyanage, N.; Ravishan, D.; Sampath, R.; et al. "Saving Precious Seconds"—A Novel Approach to Implementing a Low-Cost Earthquake Early Warning System with Node-Level Detection and Alert Generation. *Informatics* **2022**, *9*, 25. [CrossRef]
10. Wang, W.; Wang, H.; Wu, G.; Liang, X.; Chen, W.; Feng, Y. Research on the Application of SD-WAN Technology in Power Communication Scenarios. In Proceedings of the 2022 Global Conference on Robotics, Artificial Intelligence and Information Technology (GCRAIT), Chicago, IL, USA, 30–31 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 720–723.
11. Cheimaras, V.; Trigkas, A.; Papageorgas, P.; Piromalis, D.; Sofianopoulos, E. A Low-Cost Open-Source Architecture for a Digital Signage Emergency Evacuation System for Cruise Ships, Based on IoT and LTE/4G Technologies. *Future Internet* **2022**, *14*, 366. [CrossRef]
12. Kim, D.; Kim, Y.H.; Kim, K.H.; Gil, J.M. Cloud-centric and logically isolated virtual network environment based on software-defined wide area network. *Sustainability* **2017**, *9*, 2382. [CrossRef]
13. Cheimaras, V.; Peladarinos, N.; Monios, N.; Daousis, S.; Papagiakoumos, S.; Papageorgas, P.; Piromalis, D. Emergency Communication System Based on Wireless LPWAN and SD-WAN Technologies: A Hybrid Approach. *Signals* **2023**, *4*, 315–336. [CrossRef]
14. Tootaghaj, D.Z.; Ahmed, F.; Sharma, P.; Yannakakis, M. Homa: An efficient topology and route management approach in SD-WAN overlays. In Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications, Toronto, ON, Canada, 6–9 July 2020; pp. 2351–2360.
15. Duliński, Z.; Stankiewicz, R.; Rzym, G.; Wydrych, P. Dynamic traffic management for SD-WAN inter-cloud communication. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1335–1351. [CrossRef]
16. Roux, R.; Olwal, T.O.; Chowdhury, D.S. Software Defined Networking Architecture for Energy Transaction in Smart Microgrid Systems. *Energies* **2023**, *16*, 5275. [CrossRef]
17. Golani, K.; Goswami, K.; Bhatt, K.; Park, Y. Fault tolerant traffic engineering in software-defined WAN. In Proceedings of the 2018 IEEE Symposium on Computers and Communications (ISCC), Natal, Brazil, 25–28 June 2018; pp. 01205–01210.
18. Troia, S.; Sapienza, F.; Varé, L.; Maier, G. On deep reinforcement learning for traffic engineering in SD-WAN. *IEEE J. Sel. Areas Commun.* **2020**, *39*, 2198–2212. [CrossRef]
19. Ghaderi, M.; Liu, W.; Xiao, S.; Li, F. Learning Traffic Encoding Matrices for Delay-Aware Traffic Engineering in SD-WANs. In Proceedings of the NOMS 2022–2022 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 25–29 April 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–9.
20. Botta, A.; Canonico, R.; Navarro, A.; Ruggiero, S.; Ventre, G. AI-enabled SD-WAN: The case of Reinforcement Learning. In Proceedings of the 2022 IEEE Latin-American Conference on Communications (LATINCOM), Rio de Janeiro, Brazil, 30 November–2 December 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–6.
21. Borgianni, L.; Troia, S.; Adami, D.; Maier, G.; Giordano, S. From MPLS to SD-WAN to ensure QoS and QoE in cloud-based applications. In Proceedings of the 2023 IEEE 9th International Conference on Network Softwarization (NetSoft), Madrid, Spain, 19–23 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 366–369.
22. Borgianni, L.; Troia, S.; Adami, D.; Maier, G.; Giordano, S. Assessing the Efficacy of Reinforcement Learning in Enhancing Quality of Service in SD-WANs. In Proceedings of the 2023 IEEE Global Communications Conference (GLOBECOM), Kuala Lumpur, Malaysia, 4–8 December 2023; IEEE: Piscataway, NJ, USA, 2023.
23. Yang, Q.; Laurenson, D.I.; Barria, J.A. On the use of LEO satellite constellation for active network management in power distribution networks. *IEEE Trans. Smart Grid* **2012**, *3*, 1371–1381. [CrossRef]
24. Giordani, M.; Zorzi, M. Satellite communication at millimeter waves: A key enabler of the 6G era. In Proceedings of the 2020 International Conference on Computing, Networking and Communications (ICNC), Big Island, HI, USA, 17–20 February 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 383–388.
25. Borgianni, L.; Adami, D.; Giordano, S. Optimizing Network Performance and Reliability with an Integrated SD-WAN and Satellite 6G Architecture. In Proceedings of the 2023 2nd International Conference on 6G Networking (6GNet), Paris, France, 18–20 October 2023; pp. 1–4. [CrossRef]
26. De Sanctis, M.; Cianca, E.; Araniti, G.; Bisio, I.; Prasad, R. Satellite Communications Supporting Internet of Remote Things. *IEEE Internet Things J.* **2016**, *3*, 113–123. [CrossRef]
27. Holcomb, S.D.; Porter, W.K.; Ault, S.V.; Mao, G.; Wang, J. Overview on deepmind and its alphago zero ai. In Proceedings of the 2018 International Conference on Big Data and Education, Seattle, DC, USA, 10–13 December 2018; pp. 67–71.
28. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.5602.
29. Watkins, C.J.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292. [CrossRef]
30. Zhang, Z. Improved adam optimizer for deep neural networks. In Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 4–6 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–2.
31. Wunder, M.; Littman, M.L.; Babes, M. Classes of multiagent q-learning dynamics with epsilon-greedy exploration. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 1167–1174.

32. Lakhina, A.; Papagiannaki, K.; Crovella, M.; Diot, C.; Kolaczyk, E.D.; Taft, N. Structural analysis of network traffic flows. In Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems, Saint Malo, France, 26–30 June 2004; pp. 61–72.
33. Lai, K.; Baker, M. Measuring bandwidth. In Proceedings of the IEEE INFOCOM '99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No.99CH36320), New York, NY, USA, 21–25 March 1999; IEEE: Piscataway, NJ, USA, 1999; Volume 1, pp. 235–245.
34. Mora-Huiracocha, R.E.; Gallegos-Segovia, P.L.; Vintimilla-Tapia, P.E.; Bravo-Torres, J.F.; Cedillo-Elias, E.J.; Larios-Rosillo, V.M. Implementation of a SD-WAN for the interconnection of two software defined data centers. In Proceedings of the 2019 IEEE Colombian Conference on Communications and Computing (COLCOM), Barranquilla, Colombia, 5–7 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
35. Ketkar, N.; Ketkar, N. Introduction to keras. In *Deep learning with Python: A Hands-On Introduction*; Apress: Berkeley, CA, USA, 2017; pp. 97–111.
36. Dillon, J.V.; Langmore, I.; Tran, D.; Brevdo, E.; Vasudevan, S.; Moore, D.; Patton, B.; Alemi, A.; Hoffman, M.; Saurous, R.A. Tensorflow distributions. *arXiv* **2017**, arXiv:1711.10604.
37. Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; Zaremba, W. Openai gym. *arXiv* **2016**, arXiv:1606.01540.