Performance Evaluation of Overlay Networking for delay-sensitive services in SD-WAN

Sebastian Troia*†, Marco Mazzara*, Ligia Maria Moreira Zorello* and Guido Maier*†
*Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan 20133, Italy
†SWAN networks, Via Fabio Filzi 27, Milan 20124, Italy

***Abstract*—A reliable Wide Area Network (WAN) has become an imperative need for enterprises with Cloud-hosted applications and distributed branch offices. Software-Defined Wide Area Network (SD-WAN) has been regarded as the most promising technological solution for next generation enterprise networks capable of increasing network agility and reducing costs. In this paper, we present an experimental SD-WAN solution capable of running and optimizing delay-sensitive services, such as VoIP and video streaming, while minimizing downtime caused by network failures. We validate our solution thanks to two SD-WAN testbeds: the first one is deployed in a municipal network of an Italian city, while the other is emulated in our laboratory. The goal is to show the capability of SD-WAN of guaranteeing fast recovery and resilience in case of network failures, exploiting an innovative eBPF-based monitoring technique.**

***Index Terms*—Software Defined Wide Area Networking (SD-WAN), Software Defined Networking (SDN), Enterprise Networking, Traffic Engineering, Monitoring, eBPF.**

I. Introduction

Over the past decade, as enterprises have gradually shifted their applications to the Cloud, the traditional enterprise Wide Area Network (WAN) architecture had to evolve to address the new requirements of the Cloud era, such as high service availability, network resiliency and security. Enterprises use WANs to connect to their remote branches and to reach Cloud services that are provisioned by a plethora of Communication Service Providers (CSPs). As such, the rapid evolution of Enterprise networking and Information Technologies (IT) increases consistently the demand of higher capacity and higher quality WANs. CSPs are going to expand their portfolio of enterprise-focused services and, most importantly, to improve their customer products via a new emerging technology called Software-Defined WAN (SD-WAN) [1].

SD-WAN offers numerous advantages in terms of high network agility, cost savings, high availability, easier and safer management of the Enterprise Network (EN). Based on a software-defined architecture, SD-WAN delegates the control and management to a centralized controller connected uniquely to the edge devices, or Customer Premises Equipment (CPE): this means that there is no need to have direct access to the WAN internal devices (e.g. providers' routers and switches) to operate an SD-WAN system. As such, traffic engineering and monitoring applications are required for network availability and reliability.

Enterprises can orchestrate their traffic in consideration of the monitoring measurements of WAN and service performance, such as packet delay, loss, jitter, and service requirements. SD-WAN supports a new way of routing thanks to the possibility of instructing the CPEs on the basis of heterogeneous information such as: the position of the CPEs, the type of services, characteristics of the flows (TCP, UDP), etc. For instance, we could program a routing application with the goal of directly handling delay-sensitive services by improving flow latency.

In our previous work in [2], we proposed an early implementation of an SD-WAN solution based on open source components, such as OpenDaylight [3] as SDN controller and OpenvSwitch (OvS) [4] as CPEs. We presented an active monitoring application, comprised of multiple software modules running on the CPEs, with the aim of measuring WAN performance, such as packet delay and loss, to route the network traffic efficiently. In [5], we proposed different kinds of traffic engineering applications, running over the SD-WAN controller, based on traditional and Machine Learning algorithms. We proved that Machine Learning is able to anticipate the WAN failures by proactively switching critical traffic flows on a different path.

The goal of this work is to build an experimental SD-WAN solution capable of running and optimizing delay-sensitive services, such as video streaming. Unlike our previous works, in this paper we present a monitoring system able to collect not only classical network statistics acquired from the CPEs, as in [2] and [5], but also real-time transport network statistics acquired directly from the servers (or hosts) that generate the network traffic. To do this, we exploit the extended Berkeley Packet Filter (eBPF) [6]: a revolutionary technology that can run programs into the Linux kernel without changing kernel source code or loading kernel modules. Thanks to this technology, we are able to code specific programs able to monitor different parameters of transport network protocols, such as the number of TCP retransmissions of traffic flows. Furthermore, we present an application inside the ONOS SDN controller [7], called SD-WAN Traffic Engineering (TE), which interacts with our monitoring system to ensure maximum availability of delay-sensitive services by guaranteeing the resilience of the network.

Thanks to a collaboration between Politecnico di Milano, SWAN networks (a spin-off of Politecnico di Milano) and the municipal administration of an Italian town, we have developed an SD-WAN testbed and integrated our solution in their municipal network in order to connect the city hall to a remote office. Experimental results demonstrate the capability of SD-WAN to increase the overall network performance by providing fast recovery in case of link failures and transmission disruption.

In section II we show the background and motivation of this work. Section III presents the related work. Section IV provides a description of the real testbeds used to validate our implementation. Sections V and VI present the proposed SD-WAN solution and the experimental results respectively. Section VII shows the conclusion of this work.

II. Background and Motivation

Decades ago, the geographical distance between sites of enterprises was bridged by using dedicated lines leased from network operators. Usually, such leased lines had high costs and limited speeds [8]. As a consequence, many different technologies have been proposed to create the inter-site connections of ENs as overlay over public WANs, such as Asynchronous Transfer Mode (ATM), Frame Relay (FR) and Multi-Protocol Label Switching (MPLS). In particular, the most recent MPLS is currently very commonly adopted for its capability of guaranteeing Quality of Service (QoS) according to Service Level Agreements (SLAs) by setting up Label-Switched Paths (LSPs) through the IP network. On the other hand, MPLS presents some drawbacks, such as: 1) high bandwidth cost; 2) configuration complexity; 3) excessive time required to dynamically scale/upgrade the LSPs. The operational complexity of MPLS is directly related with the number of branch offices and edge devices. Although MPLS can be regarded as a milestone in EN, its high cost and complexity have recently pushed enterprises to seek an alternative solution, that is, to exploit low cost Internet/broadband services [9]. The broadband Internet access has been already used in the past in ENs by creating the inter-site overlay with Virtual Private Networks (VPNs). However, a VPN overlay is totally unreliable. SD-WAN, by introducing a centralized SDN controller, is the new EN solution able to conjugate the low cost of Internet access to a good degree of availability. The SD-WAN overlay architecture is much easier than MPLS to be dynamically configured and to adapt to the network conditions as the control and management are performed by a centralized controller. In Fig. 1 we show a typical SD-WAN solution with a controller and two Customer Premises Equipment (CPE) devices placed at the customer locations. The SD-WAN controller is responsible for routing decisions, system configuration and monitoring, while CPEs have the task of establishing secure tunnels on which to route the traffic. The tunnels can be even set up over different networks (e.g. fixed Internet and mobile). Fig. 1 shows a practical example of the operation of an SD-WAN solution. Let's assume that the packet delay on Tunnel-1 exceeds a certain threshold (10 ms) which is defined according to the QoS requirement of a delay-sensitive service. Then, the SD-WAN controller reacts by moving the traffic flow related to that service from Tunnel-1 to Tunnel-2 in order to prevent the service from being down.

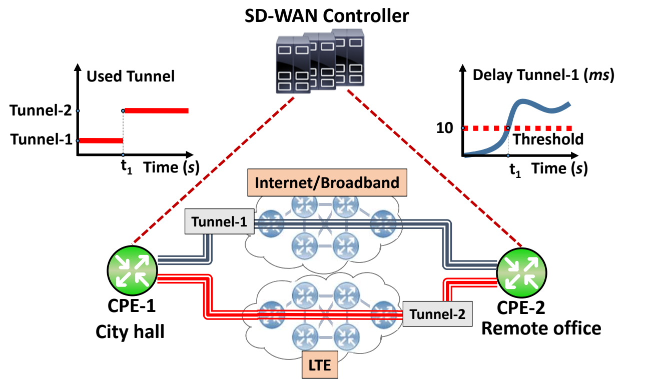

Fig. 1: Example of an SD-WAN solution composed by one controller and two CPEs.

One question arises: to what extend is SD-WAN able to actually provide availability by exploiting broadband Internet? That is the problem we would like to investigate with our work. Given the small number of independent studies we found in literature on such a topic, our opinion is that the question is still largely open. In order to fill this gap, in this work we developed an open-source SD-WAN solution based on an innovative monitoring system with the aim of guaranteeing fast recovery and network resilience in case of network failures. We performed experimental measurements through real testbeds to validate our solution.

III. Related work

The increased attention for SD-WAN across the enterprise landscape is pushing the academic research world to investigate new solutions for WAN's architecture and its management/optimization features.

Authors in [10] perform a techno-economic analysis about implementing SD-WAN with 4G/LTE for the Automated Teller Machine (ATM) networks. Most ATMs use only the Very Small Aperture Terminal (VSAT) access to connect their WAN via satellite. Having only VSAT access on most ATMs can be risky, especially if the satellite connection goes down. With SD-WAN, ATM will have at least two WAN connections to its network; as a result, if one of the connections is down, network traffic will not be interrupted. Based on the techno-economic analysis provided by the authors, the implementation of SD-WAN with 4G/LTE for the ATM network is feasible and profitable.

Phemius et al. [11] propose a Cloud network architecture in which multiple data-centers of a CSP are connected through different public Internet Service Providers (ISPs). An overlay network is created by setting up virtual tunnels. The nodes of the overlay network are the data-centers. On the control plane, they are connected to a centralized SDN controller that sets forwarding rules for the created overlay topology. Authors focus specifically on the two edge-node case, by proposing a 1:1 protection scheme with a pair of overlay tunnels. The two tunnels are created into two different WANs, managed by different ISPs. Traffic flows are divided into critical and non-critical. Whenever a failure occurs on a path, non-critical traffic is stopped, while critical flows are directed to the back-up path. The architecture is composed by different software modules working over the Java-based Floodlight Controller [12].

Z. Duliński et al. [13] propose a Dynamic Traffic Management (DTM) as a generic concept to tackle the problem of minimizing traffic transit expenses. It refers to different monetary cost of inter-WAN domain traffic, such as the one related to the network energy consumption, and other kind of costs related to the volume of traffic. The authors focus on the optimization of monetary costs related to traffic transfer via WANs. The ability of SD-WAN edge to switch traffic flows from one link to another efficiently minimizes transit expenses.

R. E. Mora-Huiracocha et al. [14], propose an SD-WAN testbed made by two data-centers and two interconnecting WANs. The goal of their work is to verify that an adequate level of QoS can be guaranteed and to provide traffic priority in an SD-WAN edge network. In the deployed scenarios, the controller can efficiently manage 300 VoIP calls, using a maximum of 16% CPU load at the edge servers.

M. Xezonaki et al. [15], introduce an SDN QoE Monitoring Framework (SQMF) able to get real time QoE monitoring metrics for maintaining the transmission network path available. SQMF measures some required metrics on the transmission path of a VoIP or video application, computes the path's QoE and changes the transmission path in case the QoE is lower than a specified threshold.

S. S. W. Lee et al. [16], design an SD-WAN edge system that can enable an enterprise to interconnect all of its geographically distributed branches through low-cost public Internet access. The whole system includes an SD-WAN controller and different applications to perform path planning, provisioning and monitoring.

Most of the research works in the literature, share the effort to demonstrate, both with experimental and simulated testbeds, the possibility of obtaining high levels of QoS for services even without QoS-guaranteed connectivity such as MPLS. However, the research works just mentioned focus mainly on the deployment and functionality of SD-WAN without investigating its performance in terms of failure recovery. With this work, we want to fill this gap by implementing an open-source SD-WAN solution on real testbeds and analysing its performance in terms of recovery time when a network failure occurs.

IV. Testbeds description

In this work, we present two SD-WAN testbeds: the first one, called Municipal testbed, is the result of a collaboration between Politecnico di Milano, SWAN networks and the municipal administration of the Italian town of Militello in Val di Catania (V.C.); while the second, called Emulated testbed, is an emulated replica of the Municipal testbed developed in our laboratory at the Politecnico di Milano. The Municipal testbed is an experimental implementation of an SD-WAN solution aimed at evaluating performance and limitations. The city-hall of Militello is connected to a remote branch office of the municipality by two different networks. In each one of the two networks a tunnel has been created to connect the two sites. The aim of the SD-WAN solution is to improve network availability by dynamically switching the inter-site traffic flows between the two tunnels. The switching occurs based on the status of the traffic flows running into the two tunnels, constantly monitored by measuring the number of TCP retransmissions.

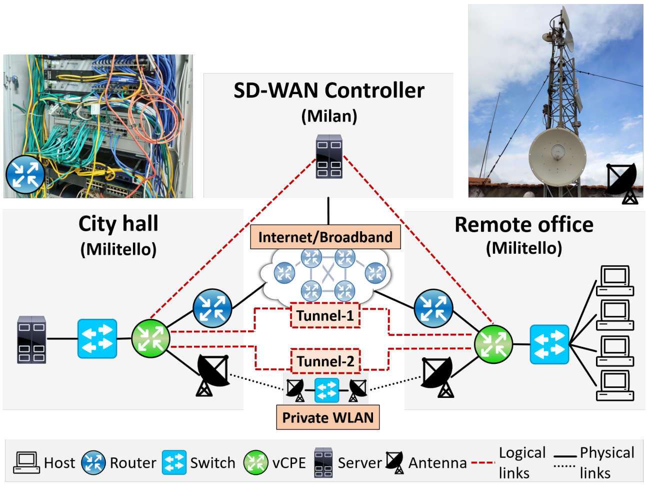

Fig. 2: SD-WAN testbed in Militello V.C.

We present the physical architecture of the testbed in Fig. 2. There are two CPEs in the data layer, one at the city hall and the other at the remote office, connecting the hosts placed in the two administrative offices to the WAN tunnels. In the testbed we have implemented the two CPEs using a pair of Raspberry Pi computers (model 3B+). The switching functionality has been obtained by installing in each Pi an OpenvSwitch (version 2.12.0). Therefore, we can refer to the two components as virtual CPEs (vCPEs). One of the two WAN networks interconnecting the vCPEs is a private WLAN owned by the municipality of Militello V.C., while the second is an Internet/Broadband network provided by an Italian ISP. We applied the Generic Routing Encapsulation (GRE) tunneling protocol [17] to implement the two tunnels, see Tunnel-1 and Tunnel-2 in Fig. 2, between the vCPEs through the two networks [18]. To implement the SD-WAN controller we used the open-source ONOS [7] SDN controller, that is responsible for: i) managing the vCPEs, issuing commands to switch traffic between the two tunnels, and ii) monitoring the performance of the two tunnels. We instantiated the controller on a server located at Politecnico di Milano (Milan, Italy) with a Linux operating system. Originally, the two sites were connected only by the WLAN, which is owned by the municipality and is therefore free of charge. However, this network is often subject to failures due to hardware problems and bad weather, which causes small displacements of the antennas from their optimal position. So, the municipality decided to contract the

ISP to switch to an interconnection through fixed broadband access, which is on the opposite a paid service. The idea underlying our testbed is, by SD-WAN, to enable a mechanism that normally routes traffic on the free-of-charge WLAN connection, switching to the ISP network only as a backup. In this way, we can improve the availability performance of the inter-site connection, while minimizing the usage of the paid ISP network, thus reducing the cost for the municipality. Our SD-WAN solution is able to manage the traffic flows between these two networks. The Emulated testbed is a replica of the Municipal testbed and it has been developed entirely within the BONSAI laboratory of the Politecnico di Milano. In this case, we used an openflow-enabled switch (Aruba JL260A 2930F-48G-4SFP) that is capable of creating several emulated instances of openflow switches by statically binding its physical ports to each instance. Thanks to this testbed we are able to test different traffic engineering and monitoring algorithms which will then be deployed on the Municipal testbed.

V. SD-WAN Monitoring and Traffic Engineering Implementation

We developed an ONOS application called SD-WAN TE to manage and improve traffic engineering at the edge of the municipal network. The SD-WAN TE is supported by a monitoring system responsible for getting real-time traffic flow statistics from the server placed at the city hall that runs administrative services. According to the required network performance constraints, the SD-WAN TE is in charge of switching the tunnel in uses by updating the OvS flow tables of the vCPEs. In this work, we developed a monitoring system that is based on transport layer network statistics, such as the number of retransmissions of TCP segments for each traffic flow (defined by the TCP source and destination port) running on the server. In the SD-WAN context, the network controller can only manage devices that are at the edges of the network (vCPEs). Therefore, if any link or node failures occur in the WAN networks, the effect is reflected on the TCP traffic in the form of increasing number of retransmitted segments. Our Monitoring System (MS) is in charge of measuring the number of TCP retransmissions of the traffic flows and warning the network controller of possible failures on the specific WAN. The MS will trace TCP retransmissions on the server by using eBPF [6].

Considering Fig. 3, we implement a Monitoring System (MS) able to collect different kind of information from the traffic flows that are running into the server in real-time, such as: IP source and destination addresses, TCP source and destination ports. Specifically, it counts the number of TCP retransmissions per traffic flow in a given time interval and monitors a pre-defined threshold (TH) is exceeded. TH represents the maximum number of TCP retransmissions for each traffic flow. TH can be set directly by the user as a parameter of the MS. If a traffic flow overrun the threshold, the host will send out an Alert Packet (AP) containing the information of that connection. The SD-WAN TE application will read the payload of the AP and will install the rules on the vCPEs to route the traffic on a backup path using a higher priority with respect to the rules of the primary path. Considering Fig. 3, below we summarize the steps necessary for the recovery of the traffic flows:

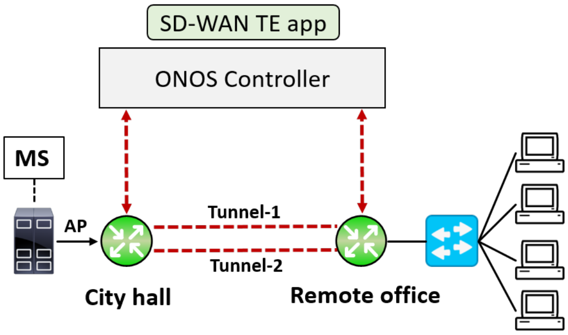

Fig. 3: SD-WAN TE and monitoring system

1) Every time a retransmission occurs, MS collects and stores the TCP traffic flows information
2) If TH is exceed by one or more TCP traffic flows, MS sends an AP to the vCPE
3) The vCPE sends this AP to the ONOS controller triggering a PacketIn event
4) The SD-WAN TE application reads the payload of the AP and triggers the switch to the Tunnel-2 for those TCP flows running on Tunnel-1

VI. Experiments and Preliminary Results

In order to measure the performance and evaluate the limitations of the proposed SD-WAN solution, we measured the total Recovery Time (RT) of multiple TCP flows affected by congestion events. We assume these TCP flows as being part of delay-sensitive application services, so with very narrow QoS thresholds, therefore, we impose a TH on the number of retransmissions equal to 10. The aforementioned congestion events are generated by inducing packet loss on both tunnels for each testbed. We used NetEm [19] as tool to generate the same percentage of packet loss on both testbeds, and D-ITG [20] as tool to generate network traffic.

Table I shows the total RT for the Municipal and Emulated testbed. In particular, it shows the following metrics:

- Detection Time (DT): time interval elapsed from the start of the congestion event to its detection from the MS
- Switch Path Time (SPT): time interval elapsed from the sending of the AP to the end of the congestion
- Total Recovery Time (RT): the sum of the two previous time intervals

Each metric shown in Table I represents the mean value of numerous experiments, with the same testbed setup as in section IV, with different packet loss percentages. Each experiment lasts from 1 to 10 hours with a packet loss between 1% and 10% per congestion event; while the TH is set to 10 retransmissions per second. We generate 10 TCP flows with a rate of 1000 pps that experience an average number of 50 congestion events per experiment, each of them lasting 5 seconds.

TABLE I: Recovery time comparison between E-Emulated and M-Municipal testbed

	Detection Time (sec)		Switch Path Time (sec)			Total Recovery Time (sec)	
	E	M	E	M		E	M
Packet Loss 1%	1.276	2.522	0.236	0.280	+=	1.412	2.802
Packet Loss 5%	0.247	0.565	0.151	0.287	+=	0.398	0.852
Packet Loss 10%	0.283	0.334	0.207	0.290	+=	0.490	0.624

When the packet loss is set to 5% (50 pps) and 10% (100 pps), we achieve a total RT between 0.4 and 0.5 seconds for the Emulated testbed, while an RT between 0.6 and 0.9 seconds in the Municipal one. The differences between the two testbeds regarding the DT depends on the different hardware computational capacity of the two testbeds. Instead, the SPT depends on the geographical distance between the vCPE and the SD-WAN controller, in fact it shows a lower value in the Emulated testbed while it shows a higher value in the case of the Municipal testbed because the vCPEs and the controller are located at a distance of 1000 km.

Fig. 4 shows the results of the MS in terms of bitrate and delay of a single TCP flow when it experiences a packet loss of 5%, therefore a high number of TCP retransmissions. In this example, the start and end of the congestion can be graphically identified by the drop and rise of the bitrate and by the rise and drop of the delay, respectively. The total RT span between these two time instants is around 400 ms. The vertical red line indicates the time of the 11th retransmission, that triggers the AP. In this experiment the total number of retransmissions stopped at 34, right before the switch to the backup path.

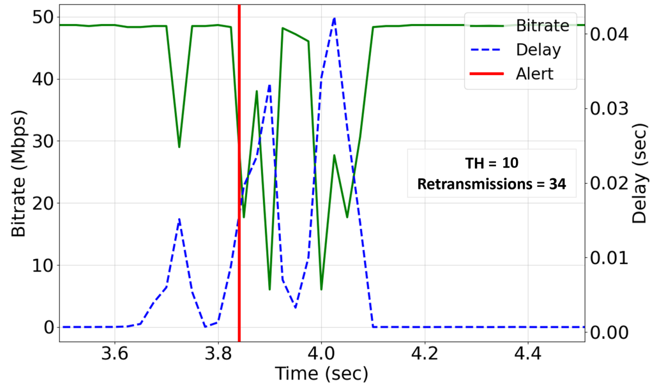

Fig. 4: Evolution of a TCP flow's Bitrate and Delay in case of packet loss and recovery

Fig. 5 shows an RT of 150 TCP flows running on the Emulated testbed with a packet loss of 5% and TH=10. The blue lines represent the time measured per flow, while the red line is the average.

The average RT is around 0.4 seconds, which is consistent with those already shown in Table I in the same conditions (packet loss=5% on the Emulated testbed) with 10 TCP flow. This means that increasing the number of TCP flows, the MS does not experience performance degradation.

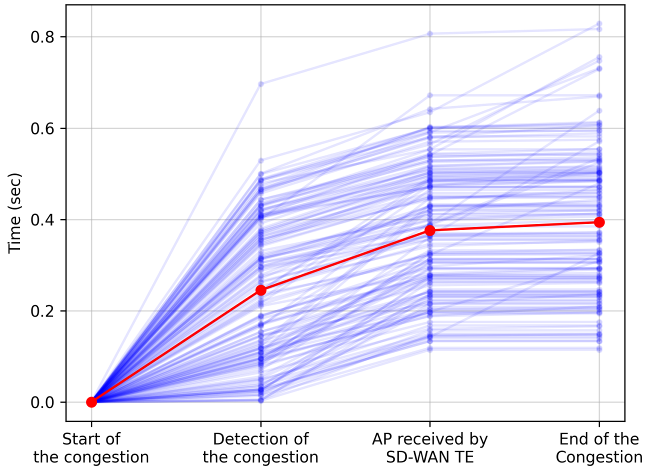

Fig. 5: Recovery time instants with 150 parallel TCP flows and 5% packet loss with the threshold TH=10 retrans/sec

VII. Conclusions

An enterprise WAN is a network that connects geographically spread sites of a company that could be located anywhere in the world. MPLS has been so far the main WAN technology for enterprise networking because of its high performance. Although MPLS has many advantages, SD-WAN is a new and fast growing paradigm that could achieve similar performance, but more cost-effectively. In this paper, we aim at evaluating the performance of an experimental SD-WAN solution deployed in two real testbeds to deliver delay-sensitive service flows with certain QoS thresholds. We have observed the advantages of SD-WAN in terms of recovery time and we have seen how this solution can provide high performance. As future work, we are going to increase the number of tunnels and the number of vCPEs to asses the scalability and the total service availability reachable by our solution.

Acknowledgment

The authors would like to thank the administration of the Municipality of Militello in Val di Catania and in particular Mario Troia, the former Head of the IT Office.

References

[1] Oracle, “Five ways sd-wan is transforming Cloud connectivity,” 2019.
[2] S. Troia, et al., ”SD-WAN: An Open-Source Implementation for Enterprise Networking Services,” ICTON, Bari, Italy, 2020, pp. 1-4.
[3] Medved, Jan, et al. ”Opendaylight: Towards a model-driven sdn controller architecture.” Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014.
[4] Pfaff, Ben, et al. ”The design and implementation of open vswitch.” Symposium on Networked Systems Design and Implementation. 2015.
[5] S. Troia, et al., ”On Deep Reinforcement Learning for Traffic Engineering in SD-WAN,” in IEEE Journal on Selected Areas in Communications, 2020.
[6] eBPF. website: https://ebpf.io/
[7] Berde, Pankaj, et al. ”ONOS: towards an open, distributed SDN OS.” Proceedings of the third workshop on Hot topics in software defined networking. 2014.

[8] R. Graziani and B. Vachon, Cisco Networking Academy: Connecting Networks Companion Guide. Cisco Press, 2014.

[9] Rangan, Raghavan Kasturi. ”Trends in SD-WAN and SDN.” CSI Transactions on ICT 8.1 (2020).

[10] S. Andromeda and D. Gunawan, ”Techno-economic Analysis from Implementing SD-WAN with 4G/LTE, A Case Study in XYZ Company,” 2020 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, 2020, pp. 345-351.

[11] Phemius, Kévin, et al. ”Implementing OpenFlow-based resilient network services.” CLOUDNET, IEEE, 2012.

[12] ”Floodlight”, October 2013, [online] Available: http://www.projectfloodlight.org/floodlight.

[13] Z. Duliński, et al., ”Dynamic Traffic Management for SD-WAN Inter-Cloud Communication,” in IEEE Journal on Selected Areas in Communications, vol. 38, no. 7, pp. 1335-1351, July 2020.

[14] R. E. Mora-Huiracocha, et al., ”Implementation of a SD-WAN for the interconnection of two software defined data centers,” 2019 IEEE Colombian Conference on Communications and Computing (COLCOM), Barranquilla, Colombia, 2019, pp. 1-6.

[15] M. Xezonaki, et al., ”An SDN QoE Monitoring Framework for VoIP and Video Applications,” 2018 IEEE 19th International Symposium on ”A World of Wireless, Mobile and Multimedia Networks” (WoWMoM), 2018, pp. 1-6, doi: 10.1109/WoWMoM.2018.8449801.

[16] Lee, Steven; et al., “Design and Implementation of an SD-WAN VPN System to Support Multipath and Multi-WAN-Hop Routing in the Public Internet”. TechRxiv. Preprint. 2020.

[17] RFC 2784, generic routing encapsulation (GRE), IETF.

[18] Wood, Michael. ”How to make SD-WAN secure.” Network Security 2017.

[19] Hemminger, Stephen. ”Network emulation with NetEm.” Linux conf au., 2005

[20] A. Botta, A. Dainotti, A. Pescapè, ”A tool for the generation of realistic network workload for emerging networking scenarios”, Computer Networks (Elsevier), 2012, Volume 56, Issue 15, pp 3531-3547.