

Resiliency in SD-WAN with eBPF Monitoring: Municipal Network and Video Streaming Use Cases

^{1,2}Sebastian Troia, ¹Marco Mazzara, ¹Ligia Maria Moreira Zorello and ^{1,2}Achille Pattavina
(1) Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan 20133, Italy
(2) SWAN networks, Via Fabio Filzi 27, Milan 20124, Italy

Abstract— A reliable Wide Area Network (WAN) has become an imperative business for enterprises with Cloud-hosted applications and distributed branch offices. Software-Defined Wide Area Networking (SD-WAN) has been regarded as the promising technological solution for next generation enterprise networks capable of increasing network agility and reducing costs. In this demonstration, we present 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 to guarantee service availability and resiliency in case of network failures, exploiting an innovative eBPF-based monitoring technique.

Keywords—SD-WAN, SDN, Network Resiliency, Enterprise Networking, Traffic Engineering, Monitoring, eBPF

I. INTRODUCTION

SD-WAN is able to conjugate the low cost of Internet access to a good degree of availability by introducing a centralized SDN controller. The SD-WAN overlay architecture is much simpler than current WAN technologies, such as Multiprotocol Label Switching (MPLS), to be dynamically configured to adapt to the network conditions. 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 performance, such as packet delay, loss, jitter, and service requirements. For both traditional and software-defined based networks, collection of traffic monitoring data can be active or passive. The former method injects additional traffic flows into the network and analyzes their behavior, while the latter does not need to inject any additional traffic since it queries network devices to obtain network statistics. In our previous work in [1], we proposed an early implementation of an SD-WAN solution based on open source components, such as OpenDaylight [2] as SDN controller and OpenvSwitch (OvS) [3] 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 [4], 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. Unlike our previous works, in this demo we present a monitoring system no longer based on network statistics acquired from the CPEs, as in [1] and [4], but on 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) [5]: a revolutionary technology that can run programs into the Linux kernel without changing kernel source code or loading kernel modules. Therefore, 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 [6], called SD-WAN Traffic Engineering (TE), which interacts with our monitoring system to guarantee network resiliency.

II. BACKGROUND ON SD-WAN

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). Technologies such as Asynchronous Transfer Mode (ATM), Frame Relay (FR) and Multi-Protocol Label Switching (MPLS) have been progressively adopted to implement the inter-site connections of ENs as overlay over public WANs. 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 to 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 CSPs 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) [7]. Based on a software-defined architecture, SD-WAN offers numerous advantages in terms of high network agility, cost savings, high availability, easier and safer management of the Enterprise Network (EN). It allows enterprises to migrate their own services to the Cloud safely by providing a flexible deployment model.

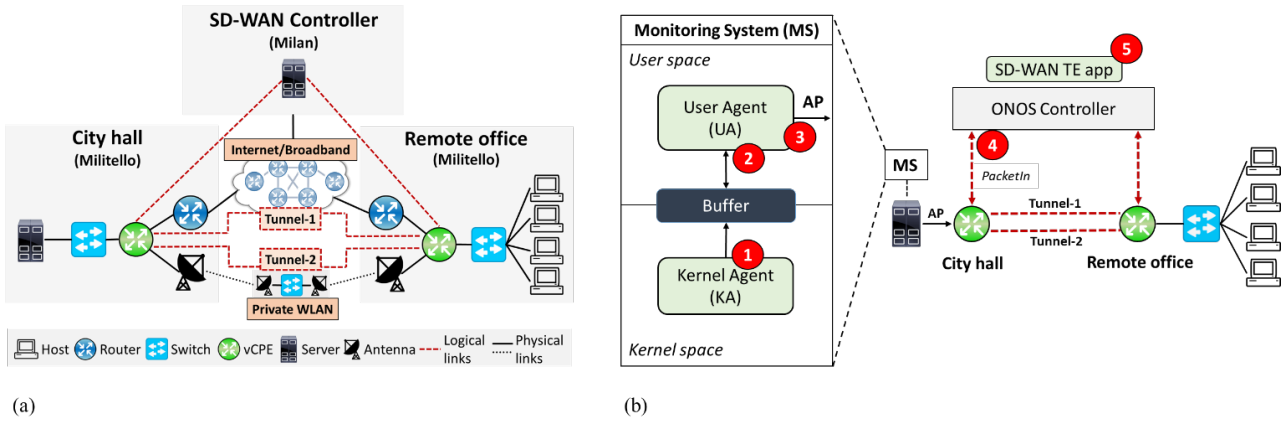


Figure 1. (a) SD-WAN testbed architecture; (b) SD-WAN TE and monitoring system

III. TESTBEDS AND USE CASES DESCRIPTION

In this demonstration, we present two SD-WAN testbeds: the first one, called Municipal testbed, is the result of a collaboration between Politecnico di Milano, SWAN networks (a university spin-off company of the Politecnico di Milano) and the municipal administration of the Italian town of Militello; 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. We present the physical architecture of the testbed in Fig. 1(a). 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, while the second is an Internet/Broadband network provided by an Italian ISP. We applied the Generic Routing Encapsulation (GRE) tunneling protocol [8] to implement the two tunnels, see Tunnel-1 and Tunnel-2 in Fig. 1(a), between the vCPEs through the two networks [9]. To implement the SD-WAN controller we used the open-source ONOS 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 switch¹ 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. The goal of this demo is to demonstrate the capability of SD-WAN to increase the overall network performance and service availability by guaranteeing network resiliency in case of link failures and transmission disruption. To do so, we designed the following use-cases to demonstrate the benefits of SD-WAN for network resiliency:

1. Municipal network use-case: the aim is to ensure that administrative services between the city hall and the branch office are restored as soon as a network failure occurs
2. Video streaming use-case: the aim is to demonstrate the effects on delay sensitive services when reconfiguration occurs to adapt to changes on network state

We use the Municipal testbed to run the first use case, while we use the Emulated testbed to run the second.

IV. 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

¹ Openflow-based switch: Aruba JL260A 2930F-48G-4SFP

switching the used tunnels by updating the OvS flow tables of the vCPEs. In this work, we developed a passive 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 [5]. Considering Fig.1(b), we implement a Kernel Agent (KA), running in the kernel space, 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. KA is triggered every time a traffic flow experiences a TCP retransmission. Afterwards, the data collected by KA is sent to a buffer, which can be read by a program in the user space. A User Agent (UA) is in charge of processing the information collected by the KA by reading the buffer. Specifically, it counts the number of TCP retransmissions per traffic flow in a given time interval and monitor whether 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 UA. If a traffic flow overrun the threshold, the host will send out an Alert Packet (AP) containing the information of that connection. The AP is crafted using Python's scapy library [10]. When the vCPE receives the AP, it issues a *PacketIn* event towards the ONOS controller by using a default miss rule. 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. 1(b), below we summarize the steps necessary for the recovery of the traffic flows:

1. Every time a retransmission occurs, KA collects and writes to the buffer the TCP traffic flows information
2. UA reads the buffer and counts the number of TCP retransmissions occurred in a given time interval
3. If TH is exceed by one or more TCP traffic flows, UA sends an AP to the vCPE
4. The vCPE sends this AP to the ONOS controller triggering a *PacketIn* event
5. 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

V. DEMO LIVE PRESENTATION

The demonstration will be done live on both testbeds. To show the practical benefits of SD-WAN, we designed the following experiments for the two use cases:

Municipal network use case (Municipal testbed):

1. We generate different TCP traffic flows [11] from the server (city hall) towards the hosts (branch office) and setup a TH=40 (arbitrarily set)

2. We artificially degrade the conditions of the tunnels (in terms of packet loss). To do so, we use NetEm [12], a software tool that allows adding losses to packets outgoing from a selected network interface
3. We show live the tunnel switching trough the ONOS web GUI and the number of TCP retransmissions for each traffic flow

Video streaming use case (Emulated testbed):

1. We generate a video streaming flow between the server and hosts running on the Tunnel-1. As for the first use case, we setup a TH=40
2. We inject additional traffic flows on Tunnel-1 in order to saturate its capacity. As a result, the video streams experience multiple TCP retransmissions
3. We show live how TCP retransmissions cause glitches in hosts that play the video. Furthermore, we demonstrate that our SD-WAN solution solves rapidly the glitches

At the end of each experiment, we show the performance of the proposed SD-WAN system in terms of recovery time, delay and total number of TCP retransmissions of traffic flows. The recovery time is defined as the time interval between the failure event and the tunnel switching.

VI. CONCLUSION

An enterprise WAN is a network that connects geographically spread sites of a company that could be located anywhere in the world. In this demo, we aim at showing the performance of two experimental SD-WAN testbeds, one in a municipal network of an Italian city and the other one in our laboratory at the Politecnico di Milano. We demonstrate the advantages of SD-WAN in terms of service availability by showing how this solution can provide network resiliency.

REFERENCES

- [1] S. Troia, et al., "SD-WAN: An Open-Source Implementation for Enterprise Networking Services," ICTON, Bari, Italy, 2020, pp. 1-4.
- [2] 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.
- [3] Pfaff, Ben, et al. "The design and implementation of open vswitch." Symposium on Networked Systems Design and Implementation. 2015.
- [4] S. Troia, et al., "On Deep Reinforcement Learning for Traffic Engineering in SD-WAN," in IEEE Journal on Selected Areas in Communications, 2020.
- [5] eBPF. website: <https://ebpf.io/>
- [6] Berde, Pankaj, et al. "ONOS: towards an open, distributed SDN OS." Proceedings of the third workshop on Hot topics in software defined networking. 2014.
- [7] Oracle, "Five ways sd-wan is transforming Cloud connectivity," 2019.
- [8] RFC 2784, generic routing encapsulation (GRE), IETF.
- [9] Wood, Michael. "How to make SD-WAN secure." Network Security 2017.
- [10] Scapy. website: <https://scapy.net/>
- [11] V. Mohan, et al, "Active and passive network measurements: a survey," International Journal of Computer Science and Information Technologies, 2011.
- [12] Hemminger, Stephen. "Network emulation with NetEm." Linux conf au., 2005

