On the Network Slicing for Enterprise Services with Hybrid SDN

Ligia M. Moreira Zorello*, Sebastian Troia*†, Serena Giannotti*, Rodolfo Alvizu*, Stefano Bregni*, and Guido Maier*†
*Politecnico di Milano, Milan, Italy. ligiamaria.moreira@polimi.it
†SWAN networks, Via Fabio Filzi 27, Milan 20124, Italy

***Abstract*—Nowadays, companies strongly rely on Virtual Private Networks (VPNs) to deliver services between geographically distributed branch offices. Internet Service Providers (ISPs) must therefore offer a reliable and cost-effective connectivity solution. VPNs are commonly based on static bandwidth allocation over MPLS tunnels, which cause over-provisioning and under-utilization of network resources. Software Defined Networking (SDN) appears as a solution to provide agile enterprise networking while reducing operators cost. This paper presents the design and implementation of a Hybrid SDN-based network application to provide dynamic services. Such an architecture enables combining centralized and distributed control with traditional VPN protocols to provision services through network slices. The application performs a flexible policy-based routing, selecting the access technology according to the Quality of Service (QoS) requirements and the network conditions. The simulations executed over an VIRL emulated environment by Cisco show that the proposed network control enables the services to be efficiently provisioned without the need of over-provisioning the resources. Furthermore, a customized network slicing over legacy equipment guarantees the service requirements.**

***Index Terms*—enterprise networking, Hybrid SDN, VPN, legacy protocol.**

I. Introduction

In the past years, companies have increasingly relied on Internet-based services, such as VoIP and video conferencing, to support their activities. These applications simplify the everyday life and the coordination of the business operations between the different sites. Consequently, reliable and efficient solutions for Wide Area Networks (WAN) are crucial. This scenario generates many challenges to Internet Service Provides (ISPs) which must handle the increasing traffic from this multitude of services over their enterprise WAN. Moreover, they must deliver flexible and cost-efficient solutions with good Quality of Service (QoS) while facing management complexity, equipment costs and scalability issues. Therefore, ISPs are required to provide a flexible yet reliable Virtual Private Network (VPN) solutions [1].

VPNs enable connecting geographically distributed sites over a public network, ensuring secure and private communication. For this, it segments the existing network into logically isolated ones. Fig. 1 exemplifies VPNs connecting the headquarters to users, partners and branch offices. In designing a VPN, many tunneling technologies can encapsulate packets over the backbone network [2]. This paper focuses on Layer 3 VPNs, which establish logical connections over IP networks.

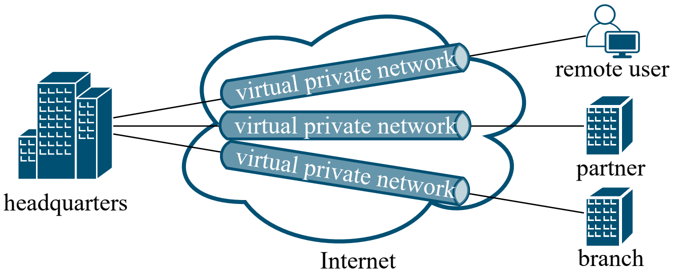

Fig. 1. Example of a virtual private network.

The technologies most widely used to support Layer 3 VPNs are based on Multiprotocol Label Switching (MPLS) and IPsec. To guarantee the applications QoS requirements, it is common practice for ISPs to statically allocate resources to meet peak-hour demands. Consequently, it over-provisions and under-utilizes WAN resources [3]. Moreover, implementing VPN over WAN involves a large number of network configurations and interactions between different protocols. For these reasons, the VPN is extremely complex to provide and maintain. ISPs observe significant impact in terms of Operating Expenses (OpEx) and Capital Expenditure (CapEx) to ensure a reliable and adaptable network architecture.

To solve the aforementioned flexibility, reliability and cost-related issues, ISPs are investing in Software-Defined Networking (SDN) [4]. This technology decouples the control and data planes, providing a better network management thanks to its programmability. Through its intelligent resource control, operators can quickly adapt to demands, improving their network efficiency. Ultimately, it enables reducing CapEx and OpEx [5]. SDNs also lead to the development of new solutions in the context of WAN such as Software Defined WAN (SD-WAN) and network slicing, attracting the interest of enterprise network managers, ISPs and network operators. SD-WAN provides automated and efficient connectivity to the WAN, and enables reducing its complexity [7]. Network slicing creates customized logical and isolated networks over the physical layer resources, such that several services are carried with different requirements [8].

This paper presents the design and implementation of a

978-1-7281-8903-1/20/$31.00 ©2020 IEEE

dynamic network slicing application based on a hybrid SDN solution that combines the private WAN and the public Internet. This tool gives flexibility in establishing services by slicing and managing the network in a centralized manner through software and combining different VPN technologies. This application was tested in an emulated environment using Cisco VIRL tool. The results show that the network control efficiently provisions services over-provisioning whilst guaranteeing the service requirements. The remaining of the paper is organized as follows. Section II describes the related works. Section III details the proposed application. Section IV explains the results and Section V concludes the paper.

II. Related Works

SDN was shown to be very effective in abstracting the control plane functions of network devices. As such, this capability can be exploited in network slicing and VPNs to guarantee the expected performance, while improving the control over the network. Consequently, coupling VPNs with SDN enhances the compliance to applications QoS and isolation of different services. Tools based on the concept of network slicing such as FlowVisor [10] were proposed in the past years. They enable slicing the hardware such that several applications can run altogether on the same physical infrastructure in an isolated manner. However, few approaches were proposed to implement VPNs making use of the SDN technology.

Lospoto *et al.* [11] describe the implementation of an SDN-based VPN that can easily manage the network and react to its dynamics. For this, they implement the VPN as protocol-agnostic exploiting OpenFlow [4] as southbound interface. However, it is difficult to combine IP headers of an encapsulated MPLS packet to the correct VPN site. The solution proposed in [12] implements OpenFlow-based MPLS VPN. It is mainly focused on traffic engineering aspects and it is bound to the traditional VNPs. Mirkhanzadeh *et al.* [13] describe an SDN-based framework relying on OpenFlow to lower the complexity when managing VPNs. With the use of SDN, this method enables users to customize service specifications. Hence, they define their own policies and connect to other entities without operator intervention.

In the context of multi-tenant networks, an SDN environment called MToS was proposed in [14]. This application runs on top of Ryu SDN framework and uses OpenFlow switches to isolate different tenant flows as a replacement of MPLS VPN. Li *et al.* also present in [15] a VPN that provides multi-tenants isolation using ONOS controller. Instead of the traditionally used southbound interface (OpenFlow), they implement Extensible Messaging and Presence Protocol (XMPP). The comparison between the two interfaces showed that XXMP presents better efficiency and scalability.

Considering SD-WAN applications, Lopez *et al.* proposed an IPsec-based solution to protect traffic exchanged over an SD-WAN [16]. They use SDN to automate the VPN configuration, which improves the performance and scalability. In the same context, Gunleifsen *et al.* describe in [17] a dynamic and automated VPN establishment for virtual network functions. They compared their Software Defined Security Associations to the IPsec-based VPNs, and showed that their application performs better in this environment.

These solutions propose a new implementation of VPNs using SDN; however, it is not feasible as they would require ISPs to entirely change their network to deploy a complete SDN architecture. In this regard, Bahnasse *et al.* describe in [18] a hybrid SDN to manage MPLS VPNs. This approach combines SDN-enabled and legacy equipment to gradually improve the network operations. Following this work, we propose a hybrid approach to build a network architecture that relies on SDN and legacy VPN protocols, *i.e.* IPsec over GRE and MPLS/BGP. This solution enables service providers to migrate to a software defined architecture gradually. Moreover, the application described in this paper also provides provisioning multiple independent slices to carry different services.

III. Dynamic Network Slicing

This section describes a dynamic and intelligent enterprise networking system to automate network configuration and optimize services performance. The proposed solution combines network slicing, SDN and legacy VPN protocols to enable a gradual deployment over an ISP network. Many tunneling technologies are available to encapsulate VPN packets over the backbone network for establishing and maintaining a logical network connection [2]. This work focuses on two Layer-3 VPN types: BGP/MPLS and IPsec over GRE.

The concept of network slicing is related to the segmentation of the physical network infrastructure in multiple logical networks. An enterprise may deliver several services between distributed branches, each one characterized by its own topology, performance and bandwidth allocation variability. For example, an organization may require a cloud service between a headquarter and a partner, and VoIP between branches. In this context, each service represents a slice that can be seen as an end-to-end logical network using shared resources, in which the end points are connected via a virtual link. The logical networks have independent control and management, and can be created or modified on demand.

Fig. 2 depicts the VPN architecture, over which multiple slices can be deployed to isolate different services. It is composed of a set of VPN routers, namely Customer Edge (CE), Provider Edges (PE) and Providers routers (P).

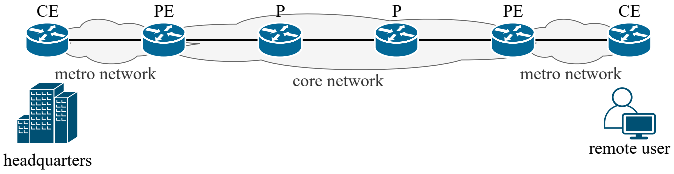

Fig. 2. Network architecture composed of CE, PE and P devices.

The CE devices represent physical or virtual routers deployed at the edge of a customer network. The PE devices are high-performance routers that connect the CE to the ISP network, and are deployed at the edge of the core network.

Finally, the P devices are core routers internally connected to multiple routers in the core of an ISP network.

A. SDN architecture

The application is based on the SDN model and combines traditional network devices and legacy protocols with the decoupling of data and control planes. Furthermore, it promotes a gradual deployment of the SDN architecture that consider new protocols and enables new network devices. Fig. 3 exemplifies this behavior when considering VPNs. The SDN architecture of this work is composed by a data plane, a hybrid control plane (distributed and centralized) and a service plane.

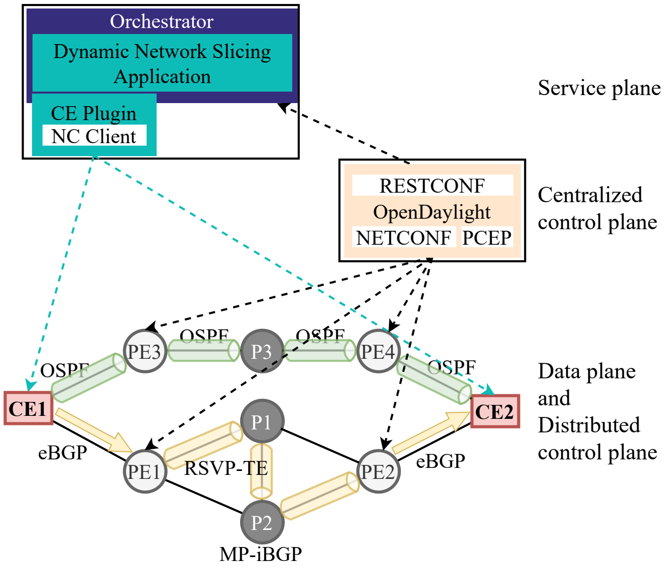

Fig. 3. Hybrid SDN architecture composed by a service plane, a centralized control plane, data plane and distributed control plane.

1) Data plane: is the part of the network that carries users traffic. It is composed by the aforementioned CE, PE and P routers, which are responsible to forward VPN packets between sites. The data plane employs MPLS and GRE tunneling protocols, whose establishment is exemplified in Fig. 3 in yellow and green, respectively.

2) Distributed control plane: refers to the set of signaling and routing legacy protocols that support the traffic forwarding in the data plane. IPsec-over-GRE VPNs do not require a signaling protocol to advertise the tunneling, and it uses uniquely OSPF as routing protocol to populate the router tables. Instead, BGP/MPLS VPN adopts Resource Reservation Protocol Traffic Engineering (RSVP-TE) as signaling protocol to create LSPs. As of routing protocols, it employs eBGP for discovering routes between CE and PE, while MP-iBGP is used for exchanging routing information between ingress and egress PEs in the core network.

3) Centralized control plane: is employed to manage network resources dynamically and in a flexible way, without having to touch individual routers of the networks. Additionally, it gives agility to the network by implementing automation and orchestration mechanisms exploiting SDN fundamental capabilities. The centralized control plane manages the PE routers to extend the set of parameters that can be managed for the slice configuration, including core network resources, such as the LSPs. In the context of this paper, we adopted OpenDayLight [19] as centralized controller.

The controller manages and obtains information from lower-layer components, *i.e.* devices at the edge of the core network, thanks to southbound interfaces. Several protocols can be used to implement this interface, and we employ Network Configuration Protocol (NETCONF) and Path Computation Element Communication Protocol (PCEP) in this work. NETCONF is based on server-client paradigm and provides mechanisms to manage and obtain configuration information from PEs. PCEP provides real-time path computation for LSPs instantiated in the core network. Another interface, namely northbound, offers an interaction between the controller and the service plane. It relies on RESTconf protocol, and enables the service plane to have visibility and control of the network devices. For this, it provides a set of common Representational State Transfer (REST) Application Programming Interfaces (APIs) to manage networking infrastructure configuration.

4) Service plane: composed by an orchestrator, a separate piece of software that runs independently of the controller. It is responsible for managing and controlling the provisioning of enterprise network slices over a WAN. It includes different software modules that optimize the network resources allocation and enables managing the underlying network in a real-time basis. The orchestrator contains an application that allocates slices over the legacy physical network to deliver enterprise services considering their specific requirements.

B. Dynamic Network Slicing Application

The main entity of this SDN architecture is the orchestrator, which manages and controls the slice provisioning. It acts as a single point of abstraction for the decisions, as it has the visibility and control of the core and edge devices. The dynamic network slicing application is responsible for the configuration of the client edge devices (CEs) and the core edge routers (PEs), described as follows.

1) Slice configuration at the edge of the customer network: performed through the CE plugin in the application by exploiting NETCONF Client (NC Client) interface. The CE plugin defines, retrieves and modifies the configuration of CE routers to configure a network slice to accommodate a service.

The dynamic slicing application performs periodic Policy-Based Routing (PRB) with VPN Routing and Forwarding (VRF)-aware services. A slice depends on several parameters that define the policies through which the routing path is selected. The parameters considered are:

- **End-points**: CEs that belong to the slice;
- **Connectivity preference**: VPN protocol desired;
- **QoS**: delay, jitter and packet loss requirement;
- **Service priority**: established based on enterprise needs;
- **Bandwidth calendar**: hourly bandwidth information.

Moreover, these requirements might change in time. By controlling the CE routers via a plugin that supports NC Client, we can dynamically change the configurations to define a slice. Thus, this software module enables network automation and zero touch provisioning of enterprise services.

In order to dynamically configure the CE devices to support network slicing, the application performs hourly updates. It starts a NETCONF session with the CE routers to obtain the CE running configurations. Then, the application updates the configuration according to the slice requirements. The network slice allocation over the WAN depends on a series of constraints related to the slice, network and service. The application solves this resource allocation problem with a heuristic executed at each hour of the day as follows.

First, the algorithm sorts the requested slices based on the priority declared by the enterprise for each service. For each slice, it checks the Service Level Agreement (SLA) of the application it should carry. Then, it searches a suitable virtual link which represents the shortest path connecting the origin and destination with a VPN protocol. If a virtual link associated to the connectivity preference and to the QoS requirements is available, the routing path of the slice is maintained. Otherwise, it searches for virtual link with a different connectivity type, *i.e.* a VPN protocol. If it is available, the virtual link is selected to the routing path. If also this virtual link is not available due to network congestion, the requested slice is refused. After optimizing policy-based routing, the application outputs the candidate configurations that are sent to the CEs to configure the requested slices.

2) Slice configuration at the edge of the core network: extends to the core network the set of capabilities for the definition and implementation of the slices. The application relies on the centralized controller for the communication to the PE devices. The controller enables the application to configure the PEs, retrieve PEs information, and create and update tunnels.

The application uses a set of REST-APIs to create an initial configuration of the PEs with the legacy protocols. The centralized controller receives the configuration from the REST-API northbound interface, converts it to an YANG module and sends it to the PEs via the NETCONF southbound interface. The application can then explore the configuration of PEs and LSPs thanks to the features provided by the PCEP southbound interface. Based on the network conditions and on the parameters requested by the slices, the application computes the LSPs and sends them to the centralized controller via the RESTconf northbound interface. The type of configuration (*e.g.* LSPs update or creation) is defined by the REST-API using HTTP POST requests with an XML-encoded body, containing the specification of PE and LSP attributes. After receiving the configuration request from the application, the centralized controller sends the path to the PEs through the southbound interface using PCEP messages. Once the PE devices retrieve the new paths, they enable the RSVP-TE protocol. It is in charge of discovering and signaling the new path creation in the data plane to adapt the network configuration to the requested slices.

IV. Results and discussion

This section presents an evaluation of the architecture proposed in Section III. The first part describes the emulation environment. Then, three simulation results are analyzed. The first aimed to evaluate the emulation environment performance. Next, we present two use cases: the dynamic slicing with a single and with multiple service slices.

A. Emulation Environment

In order to emulate the underlying infrastructure for enterprise networking, we implemented the Dynamic Network Slicing Application with Hybrid-SDN in Virtual Internet Routing Lab (VIRL) [20]. VIRL is a CISCO network virtualization and orchestration platform that enables the development of highly accurate network models. Fig. 4 shows the data plane configured in VIRL used as evaluation scenario. It includes two servers that act as hosts in the enterprise setting, two CISCO IOSv routers as CEs, four CISCO XRv routers as PEs, and three CISCO IOSv routers as P devices.

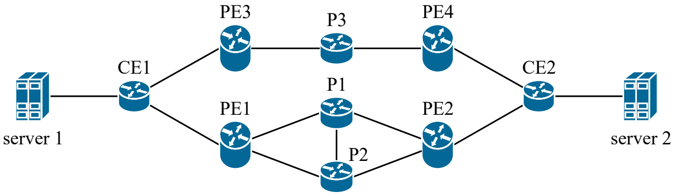

Fig. 4. Underlying infrastructure composed of two CE and three P as IOSv routers and four PE as XRv routers.

We configured a legacy Layer 3 VPN network and the SDN architecture as explained in Section III. The physical network in the data plane is controlled by the orchestrator application in the service plane, which runs on top of the OpenDayLight controller. It includes the CE plugin responsible for managing the CE configuration, implemented with the virtual Internetwork Operating System (IOSv) from VIRL. In addition, the application also controls the PE devices through the centralized controller, exploiting the southbound interfaces provided by OpenDayLight.

To analyze the network performance, we used iPerf [21] to inject traffic in the network. It is capable of measuring the maximum achievable bandwidth on IP networks and supports specifying parameters related to timing, buffers and protocols. Besides, it is possible to set client-server sessions to evaluate traffic flows. For each simulation, it reports the bandwidth, jitter and packet loss. Moreover, the traceroute tool was used to follow the path between the two servers.

B. Dynamic Allocation of a Slice

In the first use case, the enterprise requires a video-conference service between server 1 and 2, characterized by a bandwidth calendar and QoS constraints. After setting the parameters of the requested slice in the application and receiving the network congestion state as input file, the dynamic slicing application can allocate one slice for the video service following the calendar to trigger the slice changes.

Based on the service slice required parameters, the application computes the resources to be allocated every hour. This use case aims to show the performance of six path changes in the network. The set of modifications are detailed in Table I and include changes in the CE interface, modification of connectivity protocol, and path update.

TABLE I
PATH CHANGES AT THE EDGE AND CORE NODES.

Time	Endpoint	Changes at CE	Changes at PE	Path
t0	CE1	VRF-A	-	LSP 1
t1	CE1	VRF-B	-	LSP 2
t2	CE1	tunnel GRE	-	OSPF
t3	CE1	VRF-A	PE1	LSP 1
t4	-	-	PE1	LSP 2
t5	-	-	PE1	LSP 3
t6	-	-	PE1	LSP 1

In order to evaluate the bandwidth and jitter performance for this video service, an iPerf session was established between servers 1 and 2, representing the client and the server of this communication, respectively. The connection rate was set to 1 Mbit/s, its duration was 220 s, and the measurements have granularity of 1 s. In order to obtain statistical confidence, this simulation was run 10 times.

Fig. 5 and Fig. 6 depict, respectively, the bandwidth and the jitter performance obtained in this use case. These graphs highlight the information form Table I, *i.e.* the reconfiguration instants, and the regions where the changes related to the virtual links and to the network occur.

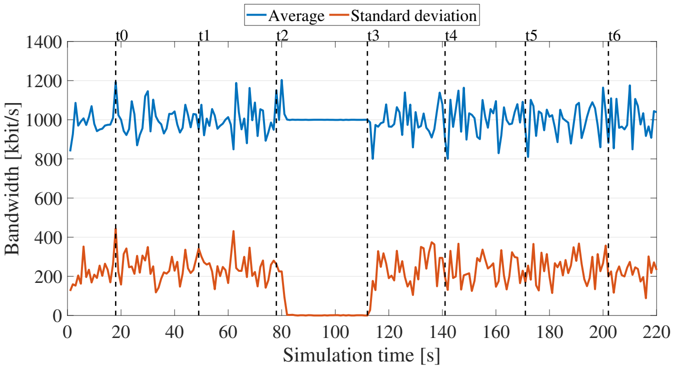

Fig. 5. Average bandwidth (blue) with standard deviation (red) during a dynamic allocation of one slice between servers 1 and 2 for video conferencing. The dashed vertical lines represent the reconfiguration instants.

These images highlight a great bandwidth variability of the MPLS/BGP VPN, while the IPsec over GRE presents much more stable results. It can be explained by the complexity of the data plane in the MPLS/BGP VPN scenario, characterized by the distributed routing protocols to manage VRFs, the LSPs signaling and the packet forwarding. Furthermore, the data plane configured by GRE tunnels is much simpler compared to the MPLS network. In particular, the network devices on the public Internet do not need any information about the VRF and the packet forwarding relies solely on OSPF.

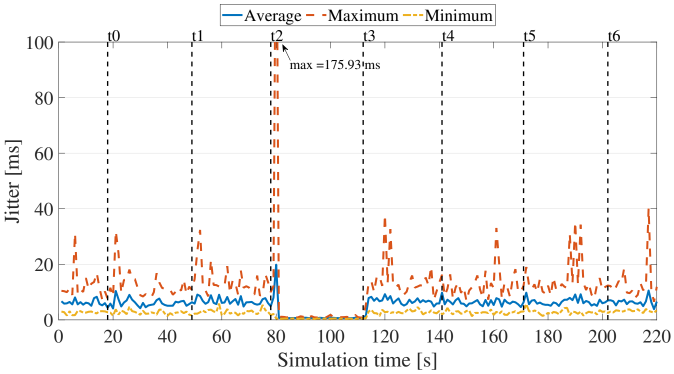

Fig. 6. Average, minimum and maximum jitter achieved during a dynamic allocation of one slice between servers 1 and 2 for video conferencing. The dashed vertical lines represent the reconfiguration instants.

Moreover, Fig. 6 also shows that there is a great jitter variability in the reconfiguration instants. This becomes more evident when the path changes due to the tunneling technology switching, *i.e.* from MPLS to GRE, taking almost 180 ms.

The jitter graph also points out another important result related to the path update. As depicted in Fig. 6 with the vertical dashed lines, the update of LSPs request occurs in the time instants 141, 171 and 201. However, the jitter only displays irregularities in the time instants 161, 191 and 221, which represent the actual moments when LSPs are updates in the core network. This delay (approximately 20 seconds) is due to the time needed for signaling the new paths of LSPs by RSVP-TE in the core of the MPLS network.

C. Dynamic Allocation of Multiple Slices

This section demonstrates the flexibility in the network resources allocation provided by the Dynamic Network Slicing application, according to the business needs.

For this use case, the enterprise requires three services: mission-critical data, VoIP and video streaming and decides to allocate different bandwidth values for each service in specific hour, according to its needs. Each service is also characterized by particular, as described in Table II.

TABLE II
BANDWIDTH CALENDAR AND QOS CONSTRAINTS RELATED TO SERVICES.

Service	Bandwidth calendar [kbit/s]				Delay	Loss	Jitter
	t0	t1	t2	t3	[ms]	[%]	[ms]
Mission-critical	25	100	100	100	300	2	150
VoIP	25	50	50	25	100	1	100
Video	100	100	100	100	4000	5	300

To accommodate the requested slices, the application evaluates the mapping of resources at each time slot and updates the slice allocation to be compliant with the requests. Based on the parameters previously described, the application updates the paths as shown in Table III.

To simulate the traffic flows on the physical network, server 1 starts three iPerf sessions with server 2, each dedicated to a service. The iPerf connections use a UDP traffic flow with

TABLE III
CHANGES IN THE EDGE AND CORE NETWORKS.

Services	t0	t1	t2	t3
Mission-critical	VRF-A	VRF-A	VRF-A	VRF-A
VoIP	VRF-B	GRE	GRE	VRF-B
Video	GRE	GRE	GRE	GRE

distinct ports for each service. The mission-critical and video streaming services have a data rate of 100 kbit/s, while 50 kbit/s is set for VoIP. A total of 10 simulations is run, each one lasting 130 s with measurements granularity of 1 s.

Fig. 7 illustrates the instants in which the reconfiguration occurs and the relative bandwidth value reached.

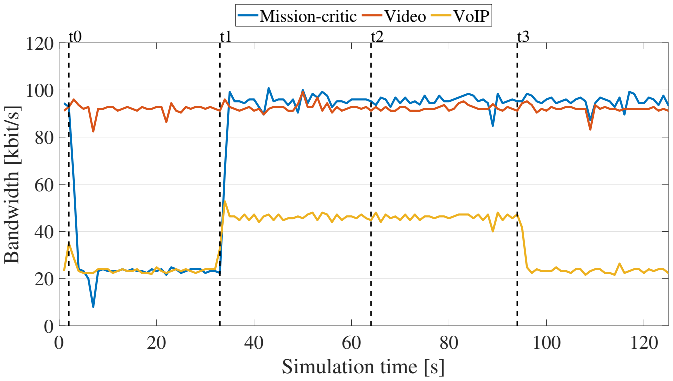

Fig. 7. Bandwidth allocation for mission-critical (blue), video streaming (red) and VoIP (yellow) data service between servers 1 and 2.

The chart demonstrates that, by controlling the CE devices in the data plane, the Dynamic Network Slicing application is able to assign the required bandwidth to each service that flows on the physical network in each reconfiguration instant. In addition, it shows that the modification of virtual links due to the change of connectivity technology does not affect significantly the bandwidth allocation, as it is fairly stable along all simulation. Consequently, the Dynamic Network Slicing application in the controller enables flexibility when controlling the hourly bandwidth allocation without the need to over provision the resources.

V. FINAL REMARKS

This paper describes the design and implementation of a hybrid SDN architecture that supports dynamic allocation of slices in an enterprise WAN. The solution enables progressively deploying an automated resource allocation in ISP networks by combining SDN and legacy protocols such as MPLS/BPS and IPsec over GRE. The deployment of a policy-based routing over different access technologies guarantees flexibility in the network resource allocation. At the same time, it avoids over-provisioning and under-utilizing static tunnels as the traditional enterprise VPNs. In addition, thanks to the VRF-aware services, it was possible to obtain a customized network slicing that guarantees meeting QoS and bandwidth requirements for traffic flows of different applications.

Further work will focus on implementing a monitoring software module to enable having a real-time network performance evaluation and linear programming to compute the optimal resource allocation of the underlying physical network. Moreover, machine learning techniques could be exploited to implement network slicing over physical network infrastructures composed of multiple independent ISPs [22].

ACKNOWLEDGMENT

The work leading to these results has been supported by SWAN Networks, a spin-off company of Politecnico di Milano.

REFERENCES

[1] H. Lee *et al.*, ”End-to-end QoS architecture for VPNs: MPLS VPN deployment in a backbone network,” *ICPP*, Toronto, Canada, 2000.

[2] T. Lackorzynski *et al.*, ”A Comparative Study on Virtual Private Networks for Future Industrial Communication Systems,” *IEEE WFCS*, Sundsvall, Sweden, 2019.

[3] M. A. Ridwan *et al.*, ”Recent trends in MPLS networks: technologies, applications and challenges,” *IET Communications*, vol. 14, no. 2, pp. 177–185, 2020.

[4] N. McKeown *et al.*, ”OpenFlow: Enabling innovation in campus networks,” *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[5] D. Kreutz *et al.*, ”Software-Defined Networking: A Comprehensive Survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[6] Z. Yang *et al.*, ”Software-Defined Wide Area Network (SD-WAN): Architecture, Advances and Opportunities,” *ICCN*, Valencia, Spain, 2019.

[7] R. Alvizu *et al.*, ”Comprehensive survey on T-SDN: Software-defined Networking for Transport Networks,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2232–2283, 2017.

[8] T. Soenen *et al.*, ”Demystifying network slicing: From theory to practice,” *IFIP/IEEE IM*, Lisbon, Portugal, 2017.

[9] S. Patil, M. S. Subhedar, ”Analysing MPLS Performance by SDN,” *Advances in Intelligent Systems and Computing*, vol. 814, 2019.

[10] R. Sherwood *et al.*, ”Can the production network be the testbed?,” *USENIX OSDI*, Vancouver, Canada, 2010.

[11] G. Lospoto *et al.*, ”Making MPLS VPNs manageable through the adoption of SDN,” *IFIP/IEEE IM*, Ottawa, ON, Canada, 2015, pp. 1155–1156.

[12] A. R. Sharafat *et al.*, ”MPLS-TE and MPLS VPNS with openflow,” *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 452–453, 2011.

[13] B. Mirkhanzadeh *et al.*, ”SDxVPN: A software-defined solution for VPN service providers,” *IEEE/IFIP NOMS*, Istanbul, Turkey, 2016, pp. 180–188.

[14] W. Jia *et al.*, ”MToS: Multi-Tenant Network Over Software Defined Networking,” *IEEE SOCA*, Kaohsiung, Taiwan, 2019.

[15] Y. Li *et al.*, ”Enabling Multi-Tenants Isolation for Software-Defined Cloud Networks via XMPP and BGP: Implementation and Evaluation,” *FiCloud*, Istanbul, Turkey, 2019.

[16] G. Lopez-Millan *et al.*, ”Towards a standard SDN-based IPsec management framework,” *Computer Standards & Interfaces*, vol. 66, 2019.

[17] H. Gunleifsen *et al.*, ”Dynamic setup of IPsec VPNs in service function chaining,” *Computer Networks*, vol. 160, pp. 77–91, 2019.

[18] A. Bahnasse *et al.*, ”Smart Hybrid SDN Approach for MPLS VPN Management and Adaptive Multipath Optimal Routing,” *Wireless Personal Communications*, 2020.

[19] J. Medved *et al.*, ”OpenDaylight: Towards a Model-Driven SDN Controller architecture,” *IEEE WoWMoM*, Sydney, Australia, 2014.

[20] J. Obstfeld *et al.*, ”VIRL: The Virtual Internet Routing Lab,” *ACM Conference on SIGCOMM*, Chicago, IL, USA, 2014.

[21] J. Dugan *et al.*, ”iPerf - The ultimate speed test tool for TCP, UDP and SCTP”, [Online]. Available: https://iperf.fr/

[22] D. Andreoletti *et al.*, ”A Privacy-Preserving Reinforcement Learning Algorithm for Multi-Domain Virtual Network Embedding,” in *IEEE Transactions on Network and Service Management*, 2020.