SD-WAN for Cloud Edge Computing Continuum interconnection

Ayoub MOKHTARI, Adlen KSENTINI
Eurecom Biot, France,
mokhtari@eurecom.fr ksentini@eurecom.fr

***Abstract*—Driven by the emergence of IoT services and the need to process data as close as possible to its source, the shift from centralized Cloud computing to Cloud-Edge computing has become essential. This transition has given rise to a new computing paradigm known as the Cloud Edge Computing Continuum (CECC), where microservices that constitute Cloud-native applications are geographically dispersed across different federated CECC nodes. This transformation has brought with it new challenges in terms of interconnecting geographically dispersed CECC nodes through Wide Area Networks (WANs) to support microservices interconnection needs (bandwidth, latency, etc.). To address this challenge, in this paper, we leverage Software Defined Wide Area Networks (SD-WAN) for interconnecting the CECC nodes. We first introduce a comprehensive CECC SD-WAN architecture by demonstrating how dynamic application-aware routing facilitated by SD-WAN can overcome these limitations, ensuring efficient and flexible network performance across geographically dispersed Cloud sites. Then, for the instantiation of the architecture, we adapt an open-source SD-WAN implementation to the context of CECC. We also show performance improvement, with our approach being 10% more efficient in terms of memory usage than the VRF lite approach in the open-source implementation.**

***Index Terms*—SD-WAN, SDN, Cloud Edge Computing Continuum.**

I. Introduction

The next evolution of Cloud computing is the extension of the centralized computing architecture towards the Edge and Far Edge, introducing the notion of Cloud Edge Computing Continuum (CECC). By Far Edge, we mean nodes like IoT devices, gateways, UAV, etc., which can provide computation capabilities. Meanwhile, applications are developed with the spirit of microservices or Cloud-native instead of a monolithic block application. In the era of CECC, the different microservices constituting the application will be deployed separately, usually as containers, such as some microservices being deployed at the central Cloud while others at the Edge or Far Edge. The placement of microservices obeys a logic that depends on the application's requested QoS support. For instance, for low latency requirements, critical microservices should be deployed at the Edge or Far Edge (i.e., end-user or IoT devices). Therefore, the network connectivity between the different microservices composing the application is very critical. In this context, the interconnection of the different components of the Cloud-Edge continuum nodes is very important and critical. The interconnection of CECC nodes (i.e., centralized Cloud, Edge, and Far Edge) can be envisioned at two levels. The first level concerns the control plan, which needs to interconnect the CECC nodes. These links do not need specific treatment and can rely on Internet connectivity or Virtual Private Network (VPN) links using MPLS or IPSec. However, the data plan links that interconnect the different CECC nodes carrying the data plane of the deployed service function chain require (i) programmability to allow the network orchestrator, for instance, to specify the level of QoS [1] and the level of resiliency that needs to be supported for each deployed service; (ii) enforcing QoS to guarantee SLA of some services, such as low-latency, high bandwidth, and low-loss rate ; (iii) Resiliency support by being able to select different routes for the same service. To satisfy the requirements mentioned above, two approaches can be envisioned for the interconnection of the CECC nodes. The first approach assumes that all CECC nodes can control the underlying network to specify the level of QoS and path selection for each deployed service function chain. For example, this could involve building Segment Routing (SRv6) routes or MPLS tunnels. Typically, Cloud operators have this capability when interconnecting their own Data Centers using multiple MPLS tunnels, where several L2 tunnels (such as VXLAN or GENEVE) are multiplexed to connect services running as containers or Virtual Machines (VMs). However, such an approach cannot be envisioned as the Cloud and Edge providers use network links operated by third-tier Internet Service Providers (ISP). Usually, CECC includes computing resources from the Edge and Far Edge, which are connected using classical network connectivity and in most cases, there is no control over the underlying network connectivity.

The second approach is to use overlay networks over the existing Internet connectivity link to interconnect the different CECC nodes. Among the most prominent and well-adapted solutions to CECC is based on Software-Defined WAN (SD-WAN). SD-WAN is an architecture that leverages Software Defined Network (SDN) principles and aims at simplifying the management and operation of the networks (with a particular focus on WAN scenarios) by decoupling the networking hardware from its control programs and using software and open APIs to abstract the infrastructure and manage the connectivity and services. SD-WAN is able to create overlay networks on top of heterogeneous underlay networks, as well as from different ISPs, supporting multiple WAN connections concurrently. The various WANs can have different performance levels and costs, as seen, for example, in the cases of the Internet, MPLS, 4G/5G,

and others.

The Overlay Network created by SD-WAN allows having a dynamic topology (full-mesh/hub-and-spoke) constituted by logical links between different SD-WAN edges, enabling new paradigms application-aware, policy-driven, and orchestrated connectivity between SD-WAN users. As described in [2], SD-WAN can interconnect sites using different topologies, including Mesh, which will allow building a full overlay, where routing can be envisioned between the different SD-WAN Edge nodes supporting application SLA, resiliency, and scalability without the need to use the underlying network resources.

Believing that SD-WAN is one of the key technologies to interconnect the CECC nodes as it satisfies the three above-mentioned requirements, we propose in this paper a high-level architecture of CECC SD-WAN, featuring an implementation of application-aware routing leveraging Linux networking as well as an adaptation of an open-source implementation to the context of CECC.

The rest of the paper is structured as follows. Section II provides a background and high-level overview of the CECC. Then, it highlights some related works on SD-WANs, from ones focusing on only private WAN to those on enterprise networking and Cloud to open source ones. Section III describes the overall CECC SD-WAN architecture. Section IV presents the architecture instantiation leveraging an open source SD-WAN, as well as the contributions to this solution enabling an application-driven routing and hybrid WAN. Section V discusses the evaluation experiments comparing the open-source VRF approach to our Linux Routing Table and mangle table-based approach. Finally, Section VI concludes the paper.

II. Background

Despite the simplicity and ease of development offered by the monolithic model, it presented limitations in scalability and maintenance as applications grew in complexity and size. Microservices emerged as a response to these challenges, drawing inspiration from Service-Oriented Architecture (SOA) and distributed systems principles. By breaking down applications into smaller, loosely coupled services, microservices offer several advantages, including enhanced scalability, flexibility, and resilience.

The rise of Cloud and Edge computing further accelerated the adoption of microservices, providing a scalable and cost-efficient infrastructure for deploying and managing distributed systems and allowing ease of application's life cycle management. The following section introduces the CECC Framework and related works on interconnecting CECC nodes using SD-WAN.

A. Cloud Edge Computing Continuum

Figure 1 depicts an envisioned architecture design of the CECC framework. This framework considers an infrastructure where multiple computing Cloud or Edge providers can be incorporated as computing resources for microservices deployed within the CECC. The IT domain represents a computing provider infrastructure joining the CECC that can be an Edge, a Cloud or both. For instance, as shown in Figure 1, IT domain 2 integrates both Cloud and Edge resources. The framework extends to include Far Edge resources, which consist of end devices such as drones, IoT devices, user equipment, and data sources. The Edge domains are connected to the data sources and Far Edge via Edge Access Networks (EAN) like 5G or other types of access networks.

Each IT domain within this framework is governed by a local infrastructure manager, referred to as a Local Management System (LMS), responsible for managing the domain's resources and overseeing the local placement and deployment of microservices. These LMSs can range from Cloud, Edge, or Far Edge management systems to network management systems, such as SD-WAN controllers or IoT gateways. All LMSs expose APIs to the Microservices Manager and Orchestrator (MMO), enabling the deployment of microservices across the CECC infrastructures. The WAN and EAN are managed by SDN controllers, which provide interfaces for the dynamic adjustment of networking rules, enabling functionalities such as path updates when network QoS degrades and optimizing networking resources.

In addition to computing and networking resources management, the MMO is responsible for managing the application lifecycle, ensuring agility in managing the different phases of the application lifecycle and resource efficiency optimization.

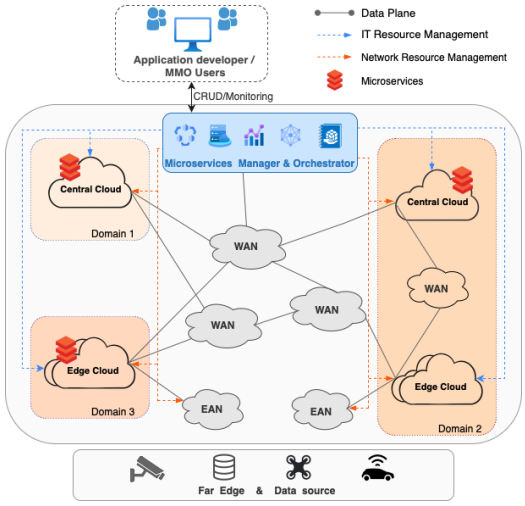

Fig. 1: CECC High-level Framework Overview

B. Related Works

To the best of our knowledge, the academic research on SD-WAN appears to be at an early stage, with very few research works done on interconnecting Cloud, Edge and Far Edge using SD-WAN. This section considers works on Cloud interconnection, enterprise networking, and works that suppose underlay network control, although we are mainly concerned with interconnecting nodes without controlling underlay networks.

In [3], Phemius et al. focus on multi-homed enclaves interconnection using OpenFlow and QoS management of applications in multi-WAN environments. Authors in [4] propose an SDN-based architecture for remote data centers interconnection aiming to improve resiliency. However, they assume that overlay

nodes are located at Internet Exchange Points (IXPs) or ISPs, or collocated with Content Delivery Network (CDN) cache nodes.

B4 [5], [6], represents the software-defined inter-datacenter backbone network deployed by Google. B4 connects data centers in different locations using a two-tier hierarchical control framework. At the lower layer, each data center site contains a network controller and hosts local control applications managing site-specific traffic. At the top layer, a logically centralized traffic engineering server is implemented. This server enforces high-level traffic engineering policies aimed to optimize bandwidth allocation between competing applications across different data center sites. In [7], the authors focus on optimizing inter-DC WAN bandwidth utilization and efficient traffic engineering, addressing the resulting congestion challenges that arise from frequent updates to meet service traffic demands. They demonstrated that leaving 10% of link capacity free allows for quick, congestion-free updates. However, these solutions are platform-centric and assume complete control over the private underlay network.

There are also open-source SD-WAN solutions like FlexiWAN [8] and EveryWAN [9]. In [8], the controller is implemented based on Free Range Routing (FRR) [10], and the Edge device or route infrastructure using FD.io Vector Packet Processor (VPP). However, it envisages a more classic approach to SD-WAN with some control functionalities still running at the virtual Edge routers. EveryWAN, on the other hand, uses a hybrid IP/SDN approach where a local control logic based on distributed IP routing coexists with a programmable IP forwarding engine controlled by a SD-WAN controller [9], leveraging Linux networking and pyroute2 netlink library.

III. Overall Architecture

The key idea of the proposed architecture is to abstract and aggregate the various networking resources, including multiple heterogeneous WAN paths, that interconnect the CECC nodes to the MMO. This approach introduces flexibility, agility, and programmability in managing the traffic flow of deployed microservices based on defined SLA/QoS requirements and current network conditions. The overall architecture is composed of three planes: management, control, and data plane, as depicted in Figure 2.

A. Management Plane

The management plan of the SD-WAN is an integrated part of the MMO. In addition to computing infrastructure like Cloud and Edge LMS exposing APIs to the MMO, the SD-WAN, acting as a LMS for the networking resources part, exposes network management functionalities to the MMO using its Northbound Interface (NBI). It bridges the gap between the abstracted networking capabilities provided by the SD-WAN controller and the operational demands of the microservices manager component of the MMO.

Given that the MMO orchestrates microservices' placement, deployment, and lifecycle management, it can provide relevant deployment configuration information to the SD-WAN controller identifying microservice traffic. This enables the controller to configure SD-WAN Edge devices with matching rules (e.g., protocol, port number, source/destination IP addresses, etc.) to identify and classify application traffic. These classes of traffic are grouped based on QoS requirements, and different microservices can be classed into one class according to their similar QoS requirements. These classes are defined at the MMO level and passed to the controller as requests to be then pushed as configurations to the Edge devices as matching rules, which will be used for traffic identification and classification.

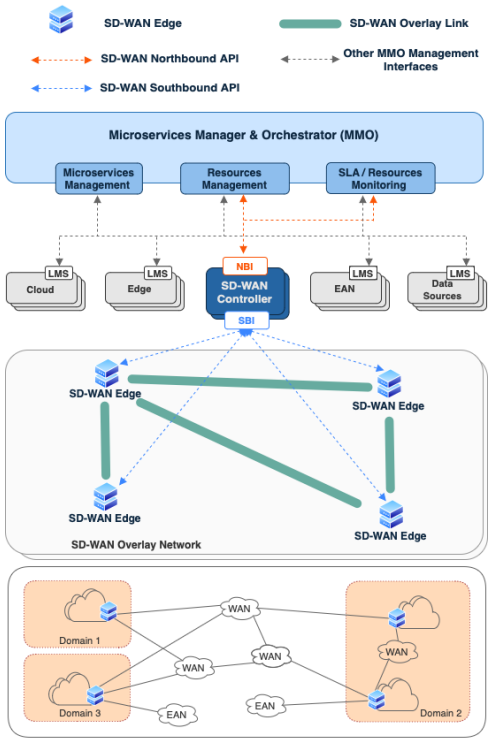

Fig. 2: Overall CECC SD-WAN Architecture

For instance, consider a microservice experiencing SLA degradation, such as exceeding the maximum acceptable latency threshold while its traffic flows through overlay network "ON-1", the MMO, assessing both the overall QoS requirements specified by the application developer and the current conditions of the overlay network leveraging a monitoring module, will request rerouting the microservice's traffic to a different overlay network "ON-2" that satisfies the QoS requirements.

Subsequently, the controller implements policy adjustments to redirect the traffic through the tunnels of the overlay network "ON-2". These traffic updates could be static, as specified by the application developer or the MMO user, or dynamic, using an integrated AI/ML-based module such as Reinforcement learning.

In case more than one overlay link satisfies the QoS requirements of a specific traffic class; the MMO can load balance traffic over overlay links that satisfy the QoS requirements of this class (e.g., per-flow load balancing). Another MMO's management plane functionality is the deployment of Edge device's Virtual Network Functions (VNFs) after adding a new CECC node.

B. Control Plane

Decoupled from the data plane, the control plane is fundamentally responsible for enforcing the control and configurations of the Edge devices. Through the NBI, the SD-WAN controller receives network update requests from the MMO, and translates them into configurations to perform at the Edge devices leveraging the Southbound Interface (SBI).

In addition to central management of the Edge devices's networking aspects, from interface configurations and overlay tunnels to routing tables and advanced policy-based routing decisions, the controller also handles the monitoring agents on Edge devices that monitor overlay tunnel-specific network metrics (e.g., latency and packet loss).

This global view of the network, in terms of different overlay topologies and real-time network performance exposed to the MMO, can be leveraged for dynamic microservices traffic management and optimizing the microservice SLA (network QoS requirements) fulfillment.

C. Data Plane

In a typical SD-WAN solution, for example, in enterprise networks, the data plane connecting different enterprise branch sites and headquarters is established by creating overlay links over both private and public IP/WAN infrastructures. However, in the CECC context, the private IP/WAN connectivity is not always guaranteed, and Cloud/Edge resources could be provided by different providers that do not share dedicated private IP/WAN links.

For instance, as shown in Figure 1, unlike IT domain 2, where computing infrastructure could be connected using private WAN, Cloud infrastructure in Domain 1 and Edge infrastructure in Domain 3 belong to different providers, hence can be connected using only public IP/WAN connectivity.

In order to connect the geographically separated CECC nodes and abstract the heterogeneous WAN connections (broadband LTE/5G, Internet, MPLS, etc.), logical overlay links are created over the existing physical underlay WAN infrastructures. The SD-WAN overlay network consists of the Edge devices (also known as CPE/vCPE in enterprise networking) and the set of logical tunnels created between these Edge devices. Each Edge device is deployed at the border of each CECC node as a gateway between the CECC node's (Cloud or Edge) local network and the different WAN; hence, representing an endpoint for overlay tunnels.

These tunnels are created using different technologies such as VXLAN [11] or GRE [12] over IPsec to encapsulate microservices traffic between CECC nodes, thus providing secure communication paths and allowing flexibility in forwarding behavior. The configurations of Edge devices for instantiating overlay tunnels, defining policies, recognizing and classifying microservices traffic for application-driven routing, and routing decisions are centrally managed at the control plane and pushed by the SD-WAN controller to the Edge devices, leveraging its SBI. In order to be able to dynamically route traffic of different microservices deployed in different CECC infrastructures, Edge devices, in addition to overlay packet encapsulation, need to be able to identify or recognize microservices traffic based on defined policies and matching rules in order to route it through a specific overlay based on defined QoS requirements or SLA agreements and network conditions, enabling an application-driven routing which is detailed in the following section.

IV. Architecture Instantiation

This section presents an implementation intsance of the proposed architecture. The architecture depicted in Figure 2 has been implemented based on the SD-WAN open source solution EveryWAN [9]. We started by leveraging the implementation of both the Edge device and the controller of EveryWAN, and the adaptation agent of the MMO for interacting with the controller NBI that was implemented in EveryWAN using gRPC [9].

In the EveryWAN solution, the SD-WAN Edge device (or so-called EveryEdge) was implemented as VNF and can be deployed on any server providing computing, storage, and network interfaces. Designed with a hybrid IP/SDN approach, the EveryEdge device combines a Programmable IP Forwarding Engine (P-IPFE), an IP routing daemon (IPRE), and an SBI also implemented using gRPC. Besides programming the forwarding entries, the controller can override the routing decisions taken by the IPRE. The IP Forwarding Engine was implemented using Linux networking, with the pyroute2 python library facilitating interactions with the Linux kernel netlink interface [9].

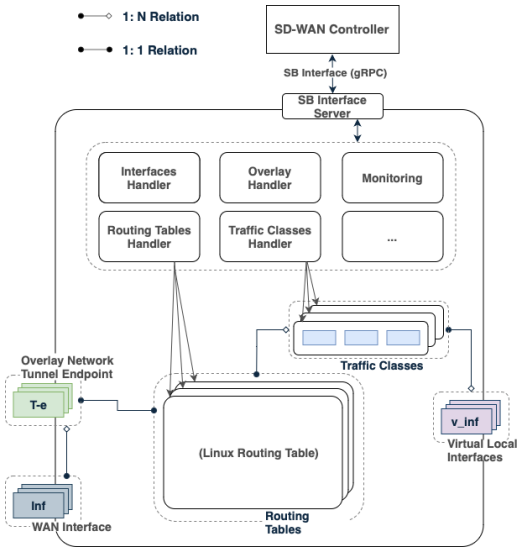

Fig. 3: Application-Driven Routing: SD-WAN Edge

For traffic redirection of a specific application over a specific overlay network, authors in [9] defined an E2E network slice in their SD-WAN solution as a combination of LAN/VLAN interfaces and an overlay network. For isolation between applications traffic in different overlay networks, a VRF lite approach was adopted mapping for each overlay network, one LAN/VLAN interface with one VRF and one physical WAN interface. However, besides the limitation of supporting only one underlay WAN interface in EveryWAN implementation, the E2E Slice definition (LAN - Overlay network - LAN) using VRF lite presents another limitation in our case. Since an interface cannot

be a member of multiple VRFs simultaneously, associating several overlay networks created on different underlay WANs with the same LAN interface was not possible with VRF lite. This limitation arises because virtual WAN interfaces and bridging would introduce complications.

To address these challenges and adapt to the CECC context, we reimplemented certain modules, like the overlay manager in the SD-WAN controller and the different handlers in the SD-WAN Edge. Additionally, we added other modules that were needed, such as application-driven routing and monitoring.

Firstly, to support hybrid WANs and enable multiple underlay links without losing Layer 3 isolation, we opted for Linux routing tables as instances of P-IPFE instead of VRF. Secondly, for the application-driven routing, as depicted in In Figure 3, each Linux routing table for an overlay network is mapped to one or more traffic classes, rather than mapping a single LAN/vLAN interface to one VRF instance. Each overlay network has its own dedicated Linux routing table, which enables logical isolation from other overlay networks and allows microservices and applications from different overlays to have distinct routing behaviors. This setup eliminates the limitation of relying solely on local interfaces to separate traffic, allowing each traffic class to have multiple ingress interfaces (LAN/VLAN interfaces).

Figure 4 illustrates the process of defining a microservice for traffic classification and routing over a specific overlay network (ON-1). As mentioned in section III the MMO has a global view of the microservices deployment, and information like deployment configurations and matching rules (e.g., protocol, port number, source/destination IP addresses, etc.) can be provided as a JSON file to the MMO network manager. The network manager uses this information to identify and classify the microservice traffic, which then requests the SD-WAN controller to add the corresponding rules for a traffic classe. At the Edge level, we used Linux IPTables tool for traffic classification and marking and IPRule for redirecting the traffic marked with the traffic class mark over the overlay routing table.

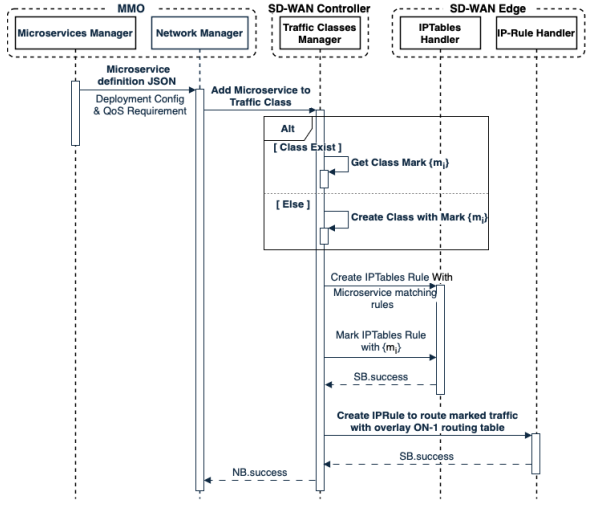

Fig. 4: Application-Driven Routing: Adding Microservice to Traffic Class

In addition, one of the main functionalities of the MMO is dynamic rerouting in response to SLA degradation, specifically violations of microservice QoS requirements. To support this, we have incorporated overlay monitoring to provide network metrics to the MMO's resource monitoring module. These metrics can be utilized by an AI-based traffic engineering module, such as a reinforcement learning algorithm, for dynamic routing. In our initial implementation, we adopted a straightforward active monitoring approach, where network measurements, including latency and packet loss, are periodically transmitted to the controller.

V. Evaluation

This section presents evaluation results on overlay creation time and Edge CPU and memory usage (VRF vs. Linux routing table).

The performance evaluations were conducted using a test bed comprising 2 Edge devices and one SD-WAN controller, each implemented as a VM. The Edge devices were connected using two different laboratory networks, and the properties of a wide area network were emulated using netem. The SD-WAN Edge VMs were allocated 2GB of RAM and 1 CPU, while the controller VM was allocated 4GB of RAM and 2 CPUs. The host machine on which the Edge device VM was provisioned was equipped with an Intel® Core™ i7-1365U processor with 10 cores, and 3.9GHz clock speed, and 32GB LPDDR5-6400MHz RAM.

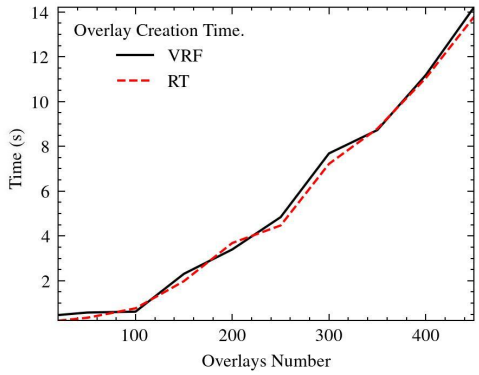

Fig. 5: Overlay Creation Time

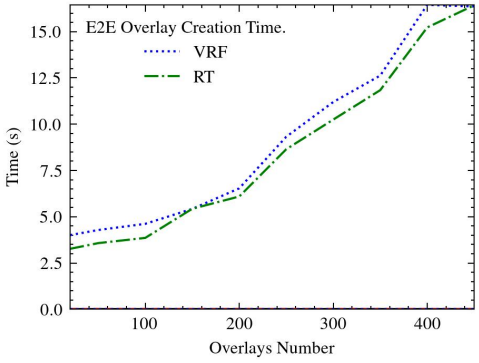

Fig. 6: E2E Overlay Creation Time

Figure 5 shows a comparison of the time required to create SD-WAN overlay tunnels in a single SD-WAN Edge between our approach using Linux Routing Table (RT) approach and the VRF approach. Initially, with overlay numbers under 100, both the VRF and RT approaches show a nearly constant time, followed by a linear increase after 100 overlay tunnels. Overall, both approaches had similar overlay creation times with minor deviations. Figure 6 represents the E2E time taken to create overlays from the MMO request moment to the overlay tunnels establishment. These results closely resemble those in Figure 5, despite the slight delay due to the laboratory network latency between the MMO/SD-WAN controller and the SD-WAN Edges.

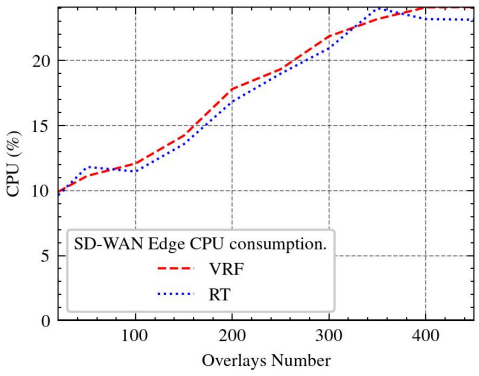

Fig. 7: SD-WAN Edge CPU Usage

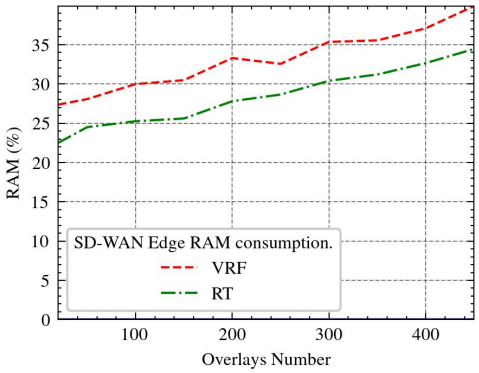

Fig. 8: SD-WAN Edge RAM Usage

The evaluation presented in Figure 7 considers the CPU usage of the SD-WAN Edge VM in terms of the overlay tunnels number created at each time in both the VRF and the RT approach. The findings show a linear rise in CPU usage as the number of overlay tunnels increases for both the VRF and the RT approach. The overall performance of both approaches is similar, although the RT approach tends to have slightly lower CPU usage.

Figure 8 illustrates memory usage for creating overlay tunnels, comparing our approach using Linux RT with the VRF approach. The findings show a linear, gradual increase in RAM usage as the number of overlay tunnels rises. Comparatively, the RT approach demonstrates greater memory efficiency, consuming approximately 10% less memory than the VRF approach. This difference is attributed to the additional overhead associated with running multiple VRF instances.

VI. Conclusion

In conclusion, this work demonstrated how SD-WAN can be integrated within the Cloud Edge Computing Continuum (CECC) to address the challenges of connecting geographically dispersed microservices and CECC nodes. We presented a high-level architecture that integrates SD-WAN with the CECC Microservice Manager and Orchestrator (MMO) and adapted an open-source SD-WAN to the CECC context, enabling hybrid WAN support to create overlay networks across multiple underlay WAN links. Additionally, we proposed an application-driven routing implementation to enable microservices traffic classification and policy-based routing. Future work will focus on enhancing dynamic application traffic engineering and optimizing QoS using AI/ML techniques.

Acknowledgment

This work was partially supported by the European Union's Horizon Research and Innovation program under AC^3 project and grant agreement No 101093129.

References

[1] K. Piamrat, A. Ksentini, C. Viho, and J.-M. Bonnin, "Qoe-aware vertical handover in wireless heterogeneous networks," in *2011 7th International Wireless Communications and Mobile Computing Conference*, 2011, pp. 95–100.

[2] P. T. A. Quang, S. Martin, J. Leguay, X. Gong, and X. Huiying, "Intent-based routing policy optimization in sd-wan," in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 4914–4919.

[3] K. Phemius and M. Bouet, "Implementing openflow-based resilient network services," in *2012 IEEE 1st International Conference on Cloud Networking (CLOUDNET)*. IEEE, 2012, pp. 212–214.

[4] A. Fressancourt and M. Gagnaire, "A sdn-based network architecture for cloud resiliency," in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*. IEEE, 2015, pp. 479–484.

[5] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu *et al.*, "B4: Experience with a globally-deployed software defined wan," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 3–14, 2013.

[6] C.-Y. Hong, S. Mandal, M. Al-Fares, M. Zhu, R. Alimi, C. Bhagat, S. Jain, J. Kaimal, S. Liang, K. Mendelev *et al.*, "B4 and after: managing hierarchy, partitioning, and asymmetry for availability and scale in google's software-defined wan," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018, pp. 74–87.

[7] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer, "Achieving high utilization with software-driven wan," in *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, 2013, pp. 15–26.

[8] (2023) Open source sd-wan and sase. [Online]. Available: https://flexiwan.com

[9] C. Scarpitta, P. L. Ventre, F. Lombardo, S. Salsano, and N. Blefari-Melazzi, "Everywan-an open source sd-wan solution," in *2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*. IEEE, 2021, pp. 1–7.

[10] (2023) Frrouting project. [Online]. Available: https://frrouting.org

[11] M. Mahalingam, D. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, and C. Wright, "Virtual extensible local area network (vxlan): A framework for overlaying virtualized layer 2 networks over layer 3 networks," Tech. Rep., 2014.

[12] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, "Generic routing encapsulation (gre)," Tech. Rep., 2000.