Converged Service Management: Wi-Fi Speed Boost

A Technical Paper prepared for SCTE by

Rahil Gandotra
Ph.D., Lead Architect
CableLabs
858, Coal Creek Circle, Louisville, CO - 80027
(303) 661-3439
r.gandotra@cablelabs.com

Yunjung Yi
Ph.D., Principal Architect & Director of Wireless Standardization
CableLabs
858 Coal Creek Circle, Louisville, CO - 80027
(303) 661-3849
y.yi@cablelabs.com

Table of Contents

Title **Page Number**

1. Introduction ... 3
2. Converged Service Management Layer ... 4
3. Wi-Fi speed boost ... 8
4. Conclusion ... 11

Abbreviations ... 13

Bibliography & References ... 13

List of Figures

Title **Page Number**

Figure 1 - High-level illustration of CSML ... 4
Figure 2 - ETSI NFV MANO architecture ... 5
Figure 3 - 3GPP-defined network slice management with interface to NFV-MANO ... 6
Figure 4 - O-RAN SMO framework ... 6
Figure 5 - High-level CSML architecture ... 7
Figure 6 - Network topology of Wi-Fi speed boost ... 9
Figure 7 - End-to-end interactions of CSML with different xNFs ... 10

1. Introduction

Two network paradigms are transforming service provider networks. First, is the transition of network functionalities being implemented as legacy physical appliances to virtualized functions running on commercial-off-the-shelf (COTS) hardware. Deploying or updating physical appliances forces operators to substantially increase both capital and operating expenditures due to the need of specialized hardware that are expensive, have high energy costs, and have limited scope for adding new functionalities [1]. Virtualizing network functions as software-based applications allows modularity and isolation of each function enabling enhanced management, network optimization, and cost reduction. Second, is the decoupling of control plane from the user plane, and the shift of closed inter-function interfaces to open standards-based interfaces. This enables the network control to be directly programmable, agile, and centrally managed, and a unified interconnection standard for white-box hardware and open-source software elements from different vendors [2].

This transformation benefits the operators by allowing for faster time to deployment, enabling enhanced flexibility, and giving them the ability to offer novel differentiated services to their users. An important requirement to realize this new paradigm is the capability to manage different access domains, such as cable and mobile, from a central management platform. While traditionally different network domains have operated in their own siloes, enabling operations convergence involves developing a common framework for deploying, configuring, and managing network functions constituting a service. Employing network domain-specific solutions with dedicated teams to provision and manage each different access service leads to a disjointed model of network management which is challenged by operational economics.

Typically, domain-specific management systems, such as for cable or mobile networks, do not have the visibility or control over other domains for fault, configuration, accounting, performance and security (FCAPS) tasks, and a central management entity can enable that inter-domain communication to achieve multi-access convergence. The Converged Service Management Layer (CSML) project aims to harmonize the management of multi-domain services by providing a single comprehensive framework to model end-to-end services and abstracting and automating the control and management of physical and virtual resources [3]. The primary goal of the project is to demonstrate the importance of converged service operations by developing novel use cases on differentiated service offerings enabled through enhanced operational agility.

CSML is envisioned to act as a master Element Management System (EMS) communicating with various domain-specific infrastructure layers, EMS, and network functions in order to develop converged services. CSML consists of a service orchestrator which leverages different workload and network orchestrators to deploy and manage services comprising of multiple functions operating in different network domains. Figure 1 illustrates the high-level concept of CSML. Different types of network functions – Physical Network Function (PNF) or Cloud-native Network Functions (CNF) – can be deployed in different zones – centralized cloud or headend or customer premises – in different network domains – cable or mobile. CSML enables the deployment of new services that converge different accesses allowing for improved quality of service for operators and seamless user experiences for subscribers.

As part of the CSML project, multiple proof-of-concepts (PoC) were developed, described in section 2, to demonstrate the converged management capability in novel use cases. This paper presents details on one use case on Wi-Fi speed boost. The motivation behind this use case was to provide the operators with the ability to incentivize their subscribers who have purchased both of their mobile and home Internet services. We extend the last PoC developed on dynamic cable speed boost by incorporating 5G core (5GC) and Wi-Fi access point (AP) into the framework to make the triggering of the speed boost more dynamic. The paper

examines the mechanisms developed for identifying mobile devices connecting to an operator's Wi-Fi network and activating cable speed boost for that device in an automated way.

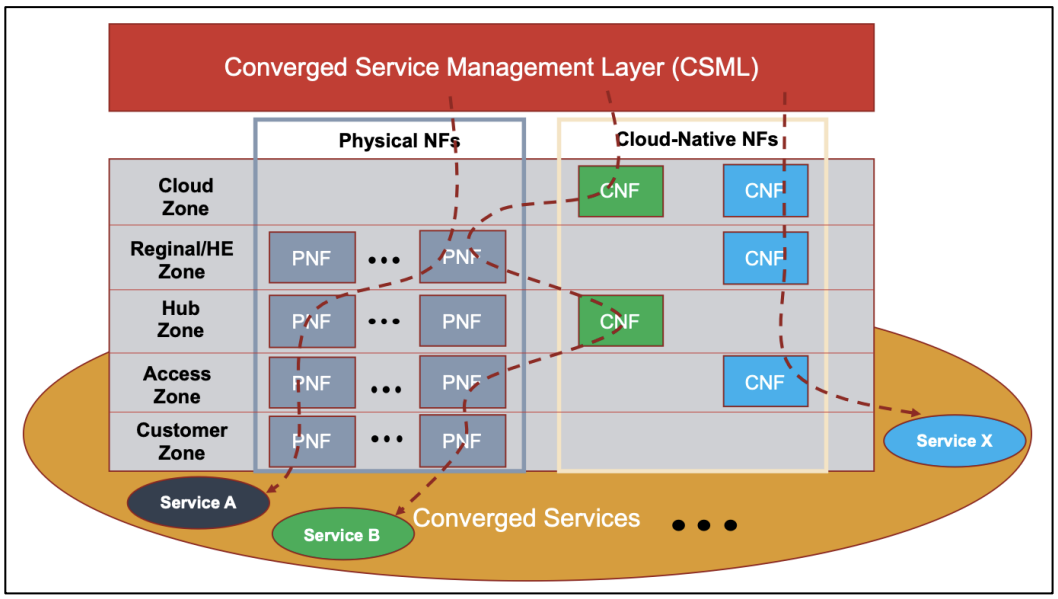

Figure 1 - High-level illustration of CSML

2. Converged Service Management Layer

Management and orchestration functionalities have been defined by multiple different standards development organizations (SDO) such as ETSI, 3GPP, and O-RAN, applicable to their domains. Certain features defined by them are independent while some are overlapping. An overarching framework that abstracts domain-specific management notions will enable a simplified approach with enhanced capabilities of cross-domain interactions without which piecemeal management solutions would need to be integrated leading to high operational complexities.

The Network Functions Virtualization Management and Orchestration (NFV-MANO) architecture specified by ETSI is shown in Fig. 2 [4]. It consists of a virtualized infrastructure manager (VIM) that is responsible for the control and management of the network functions virtualization infrastructure (NFVI) compute, storage, and network resources, usually within one operator's infrastructure domain. The Virtualized Network Function Manager (VNFM) is responsible for the lifecycle management of individual virtualized network functions (VNF). The NFV Orchestrator (NFVO) acts as the global resource manager responsible for managing services composed of multiple VNFs. Reference points with different functionalities have been specified for the interfaces between two entities. ETSI has also leveraged the NFV MANO architecture for specifying the multi-access edge (MEC) architecture in an NFV environment [4]. Interfaces between the MEC entities and MANO entities have been specified as reference points. ETSI has also developed an open-source MANO (OSM) software stack as reference implementation aligned with the NFV MANO architecture. CSML enables enhanced MANO acting as an NFVO and VNFM. It includes the capability to manage VNFs using either its built-in generic-VNFM (gVNFM) module or by interacting with external specific-VNFM (sVNFM), including an interface to the VIM layer for virtual resource management over the NFVI. Additionally, CSML can manage PNFs either via external EMS or by directly interacting with PNFs using its PNF Manager (PNFM) module over well-known management protocols.

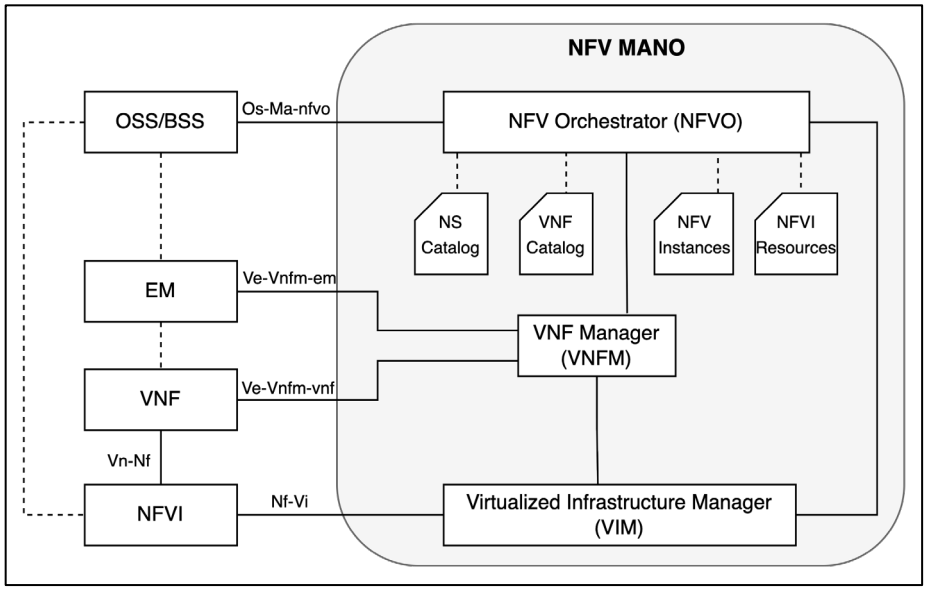

Figure 2 - ETSI NFV MANO architecture

Within the 3GPP's Technical Specification Group for Service and System Aspects (TSG SA), the working group 5 (SA5) is responsible for the management and orchestration capabilities in 3GPP systems. SA5 covers aspects such as operation, fulfillment, assurance, and automation, which includes management interactions with entities external to the network operator (e.g., service providers and verticals). The 5G management system includes features such as self-organizing networks, management data analytics, a service-based management architecture, intent-based management. Additionally, management of new 5G-enabled capabilities such as AI/ML, network slicing, URLLC, edge computing, and energy efficiency are specified. SA5 also coordinates with other SDOs such as ETSI and GSMA in the specification work pertinent to management and orchestration. For instance, Fig. 3 illustrates how the network slicing management functions defined in 3GPP utilize the ETSI NFV-MANO interfaces for configuration, fault, performance, and life-cycle management of network slices [5]. CSML architecture has the capability to incorporate such network domain-specific management functions, like the 3GPP-defined Communication Service Management Function (CSMF) and Network Slice Management Function (NSMF), to realize new use cases which help operators play a key role in catering to the needs of industry verticals apart from the traditional broadband service needs in an optimal manner.

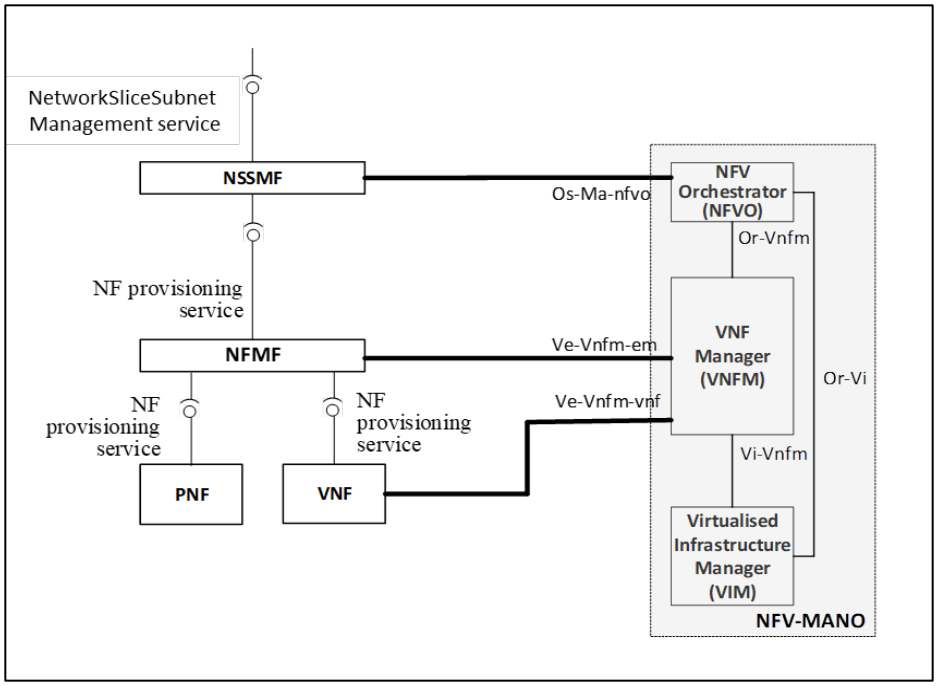

Figure 3 - 3GPP-defined network slice management with interface to NFV-MANO

The Open RAN Alliance (O-RAN) defines technical specifications and interfaces related to RAN's service management and orchestration (SMO) [6]. The SMO framework can be understood as a RAN domain controller enabling an automation platform for O-RAN radio resources. Fig. 4 depicts the SMO components and interfaces. The capabilities enabled by SMO include infrastructure management of O-Cloud, lifecycle management of the O-RAN network functions, and intelligent RAN optimization in non-real-time by providing policy-based guidance using data analytics and AI/ML models. The O-RAN O2 interface is functionally similar to the NFV-MANO Nf-Vi interface for a VIM to manage a NFVI and the O1 interface is similar to the NFV-MANO Ve-Vnfm-vnf interface for a VNMF to manage VNFs. CSML enables leveraging existing capabilities as well as incorporating new features in order to include RAN management within its purview.

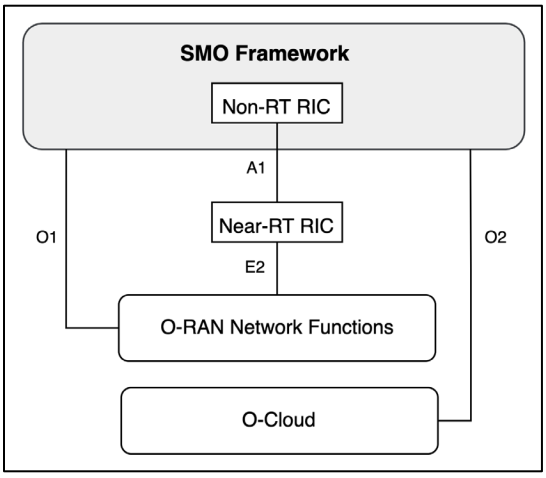

Figure 4 - O-RAN SMO framework

The primary motivation of the CSML project is to create a coherent strategy for operators to integrate these different domain-specific management functions as these technologies are introduced into their networks by enabling harmonious inter-operation of these distinct orchestration functions.

Fig. 5 illustrates the high-level architecture of CSML. The two primary frameworks, namely the design-time framework and the run-time framework, enable three types of lifecycle management (LCM) activities:

1. Service design: Allows for composing end-to-end services consisting of multiple xNFs, along with specifying their Day-0 and Day-N configuration parameters and policy rules, enabling elastic management of the service.
2. Service deployment: Involves instantiating the modeled services on to the target infrastructures (both physical and virtual) and supervising scale-in/out as needed. For deployment of VNFs and CNFs onto NFVIs, CSML interacts with the VIM APIs such as OpenStack or Kubernetes, and for provisioning of PNFs, CSML enables their plug and play by discovering the deployed PNFs and managing them using standard management protocols such as NETCONF or Ansible.
3. Service assurance: Involves monitoring the deployed services and taking closed-loop actions using analytic tools to make the framework self-healing and self-optimizing. CSML enables telemetry data collection both directly from the xNFs or from external collectors such as Prometheus or NetFlow.

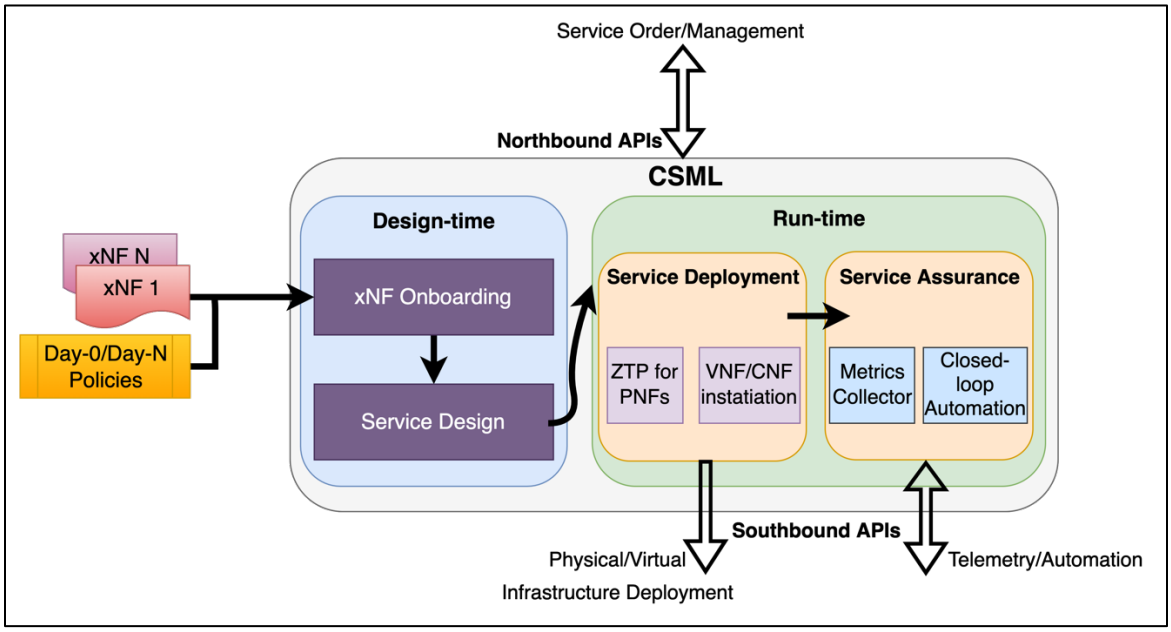

Figure 5 - High-level CSML architecture

The first use case developed during the runtime of the project was on orchestration and assurance of a wireless steering application [7]. The application consisted of eight containerized server-side functions to be deployed on three different cloud locations. CSML communicates with both public cloud and private cloud VIMs to instantiate the CNFs appropriately based on the service requirements - user-plane functions needed to be geographically closer to the user, while the control-plane functions needed to be centralized in the public cloud. Additionally, CSML enabled closed-loop automated assurance of the service by collecting near-real-time operational data, checking with pre-loaded management policies, and taking the corresponding action enabling self-healing of the service. This use case demonstrated the capability of

CSML to orchestrate CNFs-based network services on different virtualized environments by providing a common abstracted framework for intelligent deployment and management.

The second use case developed using CSML was on dynamic cable speed boost. Typically, in cable networks, the process of requesting for and implementing a change in subscriber bandwidth is manual and requires a reboot of the cable modem (CM) for the change to take effect. This process was automated by leveraging the PacketCable Multimedia (PCMM) technology that provides a mechanism to create dynamic DOCSIS service flows [8]. A virtual PCMM domain controller was onboarded into CSML to enable dynamic, closed-loop, application-specific QoS control over the legacy CMTS and CMs to support existing and future QoS-enhanced services. With this use case, CSML demonstrated how the management of physical network elements can be harmonized with virtual elements to preserve existing network investments. CSML can also act as the Telco Service Exposure Platform as part of the Linux Foundation CAMARA project enabling harmonizing of technology-specific APIs facilitating the application-to-network integration [9].

3. Wi-Fi speed boost

While the prior use cases developed using CSML focused on physical and virtual network functions orchestration, they were limited to domain-specific domains. The first use case targeted virtualized functions in the mobile and Wi-Fi domains, while the second use case focused on incorporating HFC-domain virtual and physical functions into the framework to demonstrate hybrid orchestration. The third use case, described in this section, was developed to facilitate operations convergence for operators that provide both cable and mobile services.

In order for an operator to incentivize users who have subscribed to both of their mobile and home internet services, a speed boost could be enabled for those mobile devices while connected via Wi-Fi to their cable network. This requires automated –

(i) Identification of mobile devices connecting to the Wi-Fi access points,
(ii) Verification that those mobile devices are registered with their mobile network, and
(iii) Initiating a speed boost for that device over the cable network.

To enable this, a management entity is needed to interact with multiple core and access network functions in an automated manner. CSML can act as the central orchestrator in this scenario to realize this use case by leveraging its capability to control both virtual and physical functions in mobile and cable networks.

Fig. 6 illustrates the high-level network topology of the use case implemented. CSML is based on a microservices-based architecture deployed in the public cloud in Azure Kubernetes Service (AKS). It employs Representational state transfer (REST) APIs in the southbound direction to communicate with different network functions and controllers. Interfaces were developed for CSML to interact with a 5G core, a PCMM controller, a CMTS, and a Wi-Fi AP. The 5G core was based on the open-source project free5GC [10]. The core is based on 3GPP Rel-15 standalone specifications and supports service-based interfaces. UE and RAN were simulated using open-source UERANSIM [11]. It consists of an 5G-SA UE and gNodeB implementation. The open-source OpenDaylight (ODL), a Linux Foundation Networking (LFN) software-defined controller project, was used as the PCMM controller in this use case which includes an implementation of the PCMM service as an additional module [12]. ODL was running as a virtual machine (VM) in the public cloud. ODL exposes northbound REST APIs to provision a CMTS and service flows and uses the Common Open Policy Service (COPS) protocol as transport to communicate with the CMTS. CMTS uses MAC-layer DOCSIS Dynamic Service Add/Change/Delete (DSA/DSC/DSD) messaging for service flow management for a specific CM. An Arris E6000 was used as the CMTS, a Netgear CM1000

was used as the CM, and a Flint GL-AX1800 running the OpenWrt operating system was used as the Wi-Fi AP. Another laptop was connected to the AP for throughput-testing purposes.

The design-time work involved onboarding the 5GC, RAN, and UE CNFs within CSML and composing a service. The 5GC consisted of Access and Mobility Management Function (AMF), Session Management Function (SMF), User Plane Function (UPF), Authentication Server Function (AUSF), NF Repository function (NRF), Unified Data Management (UDM), Unified Data Repository (UDR), Policy Control Function (PCF), Network Slice Selection Function (NSSF), Non-3GPP Inter-Working Function (N3IWF), and a web-UI function. CSML onboarded the 5GC, RAN, and UE CNFs packaged using Helm charts. Helm charts allow for managing complex Kubernetes applications through a collection of files that describe a related set of resources such as a virtual function, virtual network, configuration, and storage-related information. Additionally, blueprint models for ODL and CMTS were reused from the previous PoC on dynamic cable speed boost in order for their interaction with CSML. The southbound interface used to configure ODL is a REST-based environment that enables pipelining multiple API request-responses defined in a workflow, while the southbound interface used for the CMTS was a Python-based environment that allowed reusing existing scripts for managing the CMTS over SSH.

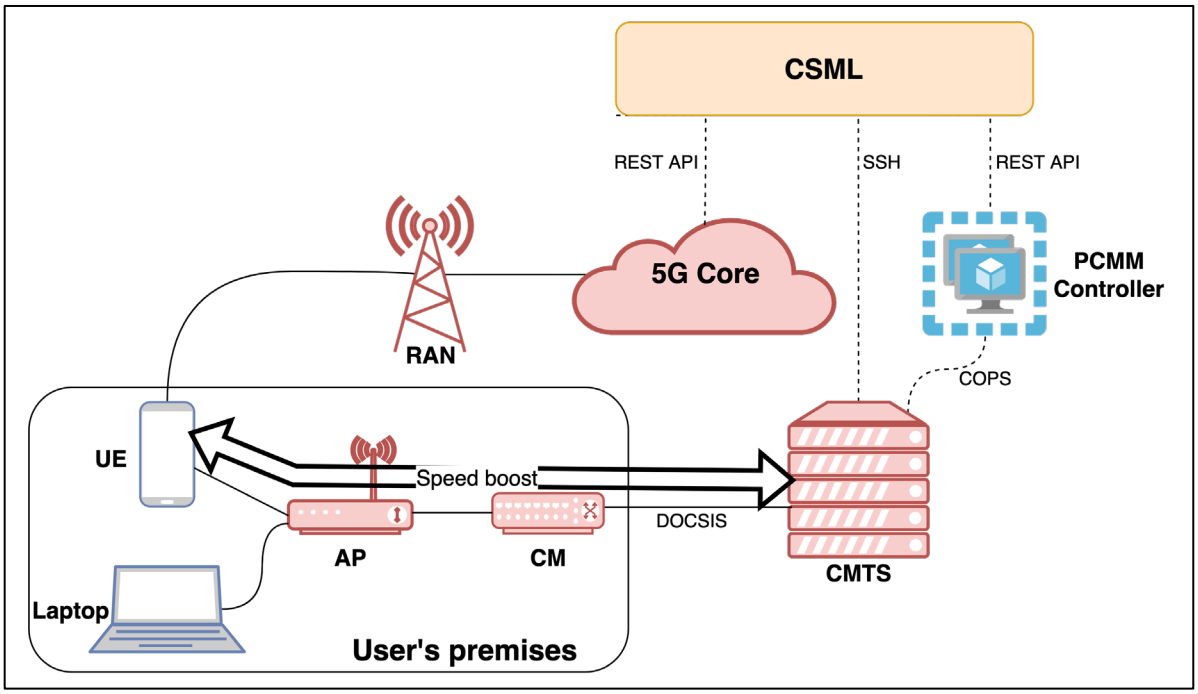

Figure 6 - Network topology of Wi-Fi speed boost

The first step in the run-time was to orchestrate the 5GC, RAN, and UE CNFs onto different Kubernetes clusters. 5GC CNFs were instantiated onto an AKS cluster, while the UE and RAN CNFs were instantiated onto a Kubernetes cluster running on a laptop acting as a mobile phone and the RAN node. This involved registering the Kubernetes clusters with CSML using cluster configuration files, employing the Kubernetes plugin within CSML to invoke Kubernetes APIs to deploy the CNFs, and configuring their interconnections [12].

To detect and identify a device connecting to the Wi-Fi AP, a mechanism was needed on the AP to interact with the device to identify if it is a mobile device, fetch its 5G identifier, and send this information to CSML. Custom scripts on the AP were employed for this purpose. A bash script was developed to identify

any new device connecting to the AP via DHCP. The script continually monitored the wireless interface on the AP for any new device connecting over Wi-Fi by tracking the MAC addresses and then obtained the IP address of the connecting device via the DHCP lease information. In case of MAC randomization, since the DHCP leased IP address changes if the MAC address changes, the mechanism on the AP to detect and obtain the current IP address for the device would suffice. Next, the AP used REST to invoke an API on the new device to fetch its International Mobile Subscriber Identity (IMSI). An API was developed to run on the UE CNF which provided the IMSI as a response to the API request from the AP. While this API on the UE was a simulated API, an operator application running on the UE could expose this information for use by the AP. For example, Android SDK has a package called TelephonyManager that provides access to information about the telephony services on the device. The getSubscriberId method within this package returns the unique subscriber ID, for example the IMSI, to a calling app that has carrier privileges.

Since the UE connects to the CM using the Wi-Fi AP, a challenge while implementing this PoC was how would the CM identify traffic from the UE, since the Wi-AP NATs all LAN IP addresses to a public IP address. With respect to the CM, the identity of the UE is hidden behind the AP, while the packet filtering and service flow enforcement for that UE's traffic will occur at the CM. The solution employed for this PoC is the AP marking all traffic matching the UE LAN IP address using a DSCP value before forwarding it to the CM, and a classifier on the CM matching that DSCP value can be used to redirect traffic from the UE to a higher-bandwidth service flow. On the AP, the mangle table of the iptables rules was used to alter the IP headers by marking all traffic matching the LAN IP address of the UE with a DSCP value, such as 56, to be used locally by the CM.

Fig. 7 depicts the multiple interactions between CSML and different xNFs. The end-to-end flow works as:

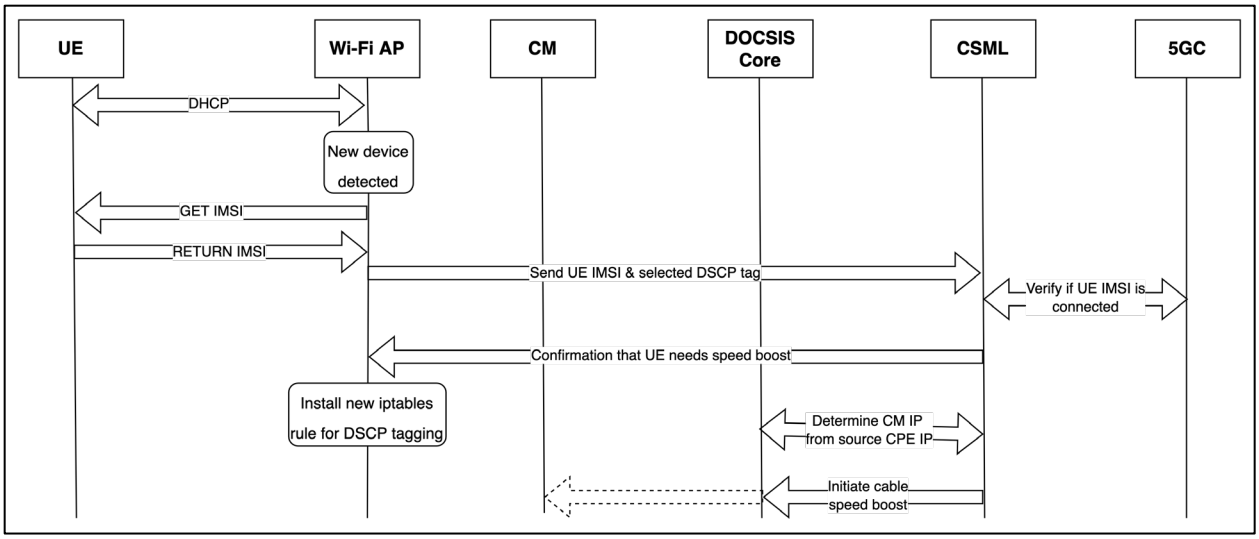

Figure 7 - End-to-end interactions of CSML with different xNFs

1. UE connects to the Wi-Fi AP using DHCP.
2. A bash scripting running on the AP detects a new device connecting over its wireless interface and acquires the LAN IP address of the UE.
3. A Python script running on the AP attempts to fetch the 5G identification, IMSI, from the UE using REST API.
4. An API running on the UE returns the IMSI to the AP. In case the device connecting to the AP is a non-mobile device, such as a laptop, the IMSI collection attempt by the AP will fail since no IMSI-exposure API will be running on the laptop.

5. Next, the AP selects a local DSCP value to be assigned to all the traffic matching the UE's LAN IP address, and sends the UE IMSI and the selected DSCP value to CSML using a REST API.
6. Once the API server running within CSML receives a request from the AP, it needs to check whether the IMSI of the UE is registered with the 5GC. It does this by invoking a REST API on the 5GC with the IMSI being sent as a payload in the HTTP request.
7. If the IMSI is registered with the 5GC using the 3GPP access, CSML then proceeds with the next steps to initiate a cable speed boost for that UE.
8. CSML first returns a confirmation to the AP that the UE requires a speed boost.
9. The AP then installs iptables rules to mark all traffic matching the UE's LAN IP address with the selected DSCP value by adding the corresponding line to the mangle table.
10. To push a PCMM policy to the correct CM, CSML needs to identify the CM IP. To do this, it uses the source IP address of the API request from the AP, connects to the CMTS over SSH, and checks which CM does that CPE IP belong to.
11. After obtaining the CM IP address, CSML sends a REST API request to ODL with the new service flow information. This request consists of the CM IP, the classifier matching all traffic from the UE's LAN IP address, and the name of the pre-configured higher-bandwidth DOCSIS service class.
12. ODL uses the COPS protocol to communicate with the CMTS about the new service flow requested, and the CMTS uses DOCSIS to push the service flow to the specific CM.
13. Once ODL returns the confirmation to CSML that the flow has been pushed, as part of verification, CSML initiates a SSH connection to the CMTS requesting the list of all service flows active for that CM IP.

The end-to-end flow described above demonstrates the level of abstraction offered by using a converged service orchestrator. Once the UE connects to the cable network through the Wi-Fi AP, the various steps needed to detect, identify, and initiate a speed boost are automated by using CSML. This enables eliminating any human intervention needed, either by the subscriber or by the operator, to achieve this. CSML enables integrations with multiple different network functions and controllers, thereby allowing for the implementation of novel use cases with the orchestrator handling much of the operational complexities associated with them.

To test the overall functioning of this use case, once the UE connects to the AP and CSML installs the higher-bandwidth service flow, throughput testing was performed on the UE and the laptop connected to the AP. While the tests performed for the UE indicated throughput up to ~450 Mbps, throughput on the laptop was capped at the subscribed bandwidth – 100 Mbps. This highlights the capability enabled by CSML for the operators to provide differentiated services to their bundled subscribers.

While the same use case could be achieved by developing various scripts to achieve individual xNF-to-xNF communication, using a centralized orchestrator provides additional benefits. Firstly, developing scripts for interaction with each vendor-specific xNF is not trivial, and the form of automation needed to achieve this without any human intervention is complex. CSML abstracts and automates various workflows to help composing an end-to-end service comprising of multiple functions and exchanges. Secondly, once a function and its associated interaction has been onboarded within CSML, the same could be used in other service designs, thereby eliminating the need to start from scratch. For example, in this PoC, the workflows defined in the previous use case on dynamic cable speed boost with ODL and CMTS were reused and incorporated with the 5GC and Wi-Fi AP to enable multi-domain management.

4. Conclusion

Network convergence allows for unifying disparate access technologies to deliver seamless connectivity enabling operators to intelligently use both mobile and fixed networks to deliver new services. To benefit

from this new paradigm, a converged service operator needs to have the ability to model end-to-end multi-domain services and to abstract and automate the control of physical and virtual network functions. This paper presented the CSML framework, highlighted the different use cases developed using it on multi-cloud orchestration, closed-loop assurance, and dynamic cable speed boost. Details were provided on the Wi-Fi speed boost PoC wherein a mobile device belonging to a subscriber of both broadband and mobile services from an operator is provided a higher broadband bandwidth. This involved detecting a mobile device connecting to the operator's AP, checking if this mobile device is registered with the operator's mobile core, and initiating a speed boost for that device over the broadband network. The paper demonstrates the value of operations convergence by demonstrating the novel capability enabled by using a centralized orchestrator for multi-domain network service comprising of different types of network functions and controllers.

In addition to the use cases described in this paper, O-RAN SMO capability was also incorporated into CSML. The O-CU, O-DU, and O-RU software was based on simulators developed by the O-RAN software community. The CNFs Helm charts were onboarded within CSML and instantiated onto a Kubernetes cluster. Configuration, fault, and performance management was implemented over the O1 interface using NETCONF and YANG models. This further augmented CSML's capability to manage a wide variety of network functions belonging to mobile and cable cores and access networks. Employing CSML in other novel use cases working with different stakeholders would help drive the adoption of the concept of converged orchestration.

Abbreviations

5GC	5th Generation mobile network core
AKS	Azure Kubernetes Service
AP	Access point
CM	cable modem
CMTS	Cable Modem Termination System
CNF	cloud-native network function
COPS	Common Open Policy Service
COTS	commercial off-the-shelf
CSML	Converged Service Management Layer
DOCSIS	Data-Over-Cable Service Interface Specification
EMS	element management system
ETSI	European Telecommunications Standards Institute
FCAPS	fault, configuration, accounting, performance, security
HFC	hybrid fiber-coaxial
IMSI	international mobile subscriber identity
MANO	management and orchestration
NF	network function
NFV	network functions virtualization
NFVI	network functions virtualization infrastructure
NFVO	network functions virtualization orchestrator
ODL	OpenDaylight
PCMM	PacketCable Multimedia
PNF	physical network function
QoS	quality of service
SDN	software-defined networking
SDO	standards development organization
VIM	virtualized infrastructure manager
VM	virtual machine
VNF	virtual network function
VNFM	virtual network function manager

Bibliography & References

[1] J.G. Herrera and J.F. Botero, "Resource allocation in NFV: A comprehensive survey," in *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, Sep. 2016, pp. 518-532.
[2] N. McKeown et al., "OpenFlow: Enabling network innovation in campus networks," in ACM SIGCOMM Computer Communications Review, vol. 38, no. 2, pp. 69-74, Mar. 2008.
[3] R. Gandotra, S. Khan, and Y. Yi, "Converged service orchestration: Dynamic cable speed boost," in *SCTE Cable-Tec Expo Fall Technical Forum*, Philadelphia, Sep. 2022.
[4] ETSI, "NFV Release 4; Management and Orchestration; Architectural Framework Specification," *ETSI GS NFV 006*, v4.4.1, Dec. 2022. [link]

[5] 3GPP, "Management and orchestration; Architecture framework," *TS 28.533*, v17.3.0, Mar. 2023. [link]
[6] O-RAN, "Architecture description," *WG1.OAD-R003*, v09.00, 2023.
[7] Cable Television Laboratories, Inc., "IWiNS—An Informed Approach to Mobile Traffic Steering", January 2021. [link]
[8] Cable Television Laboratories, Inc., "PacketCable™ Multimedia Specification", PKT-SP-MM, November 2019. [link]
[9] CAMARA: The Telco Global API Alliance, The Linux Foundation Projects, https://camaraproject.org/.

[10] free5GC, https://github.com/free5gc/free5gc.

[11] UERANSIM, https://github.com/aligungr/UERANSIM.

[12] OpenDaylight, The Linux Foundation Projects, https://www.opendaylight.org/.

[13] Edge Multi-Cluster Orchestrator (EMCO), The Linux Foundation Projects, https://project-emco.io/.

