



**UNIVERSITATEA POLITEHNICA DIN
BUCUREȘTI**



**Școala Doctorală de Electronică, Telecomunicații
și Tehnologia Informației**

Decizia nr. 694 din 28-06-2021

REZUMAT TEZĂ DE DOCTORAT

Andra-Isabela-Elena CIOBANU

**CONTRIBUȚII LA ASIGURAREA CALITĂȚII
SERVICIULUI ÎN REȚELE 5G MULTI-DOMENII ȘI
FELIATE PRIN MANAGEMENT, ORCHESTRARE ȘI
VIRTUALIZARE**

COMISIA DE DOCTORAT

Prof. Dr. Ing. Bogdan IONESCU Univ. Politehnica din București	Președinte
Prof. Dr. Ing. Eugen BORCOCI Univ. Politehnica din București	Conducător de doctorat
Prof. Dr. Ing. Virgil DOBROTĂ Univ. Tehnica din Cluj-Napoca	Referent
Prof. Dr. Ing. Sorin ZOICAN Univ. Politehnica din București	Referent
Conf.Dr. Ing. Cristian-Iulian RÎNCU Academia Tehnica Militara București	Referent

BUCUREȘTI 2021

Cuprins

Capitolul 1.....	3
Introducere	3
1.1 Prezentarea domeniului tezei de doctorat	4
1.2 Scopul tezei de doctorat.....	4
1.3 Conținutul tezei de doctorat.....	4
Capitolul 2.....	5
Managementul Orchestrării în cadrele de lucru NFV ETSI și SONATA.....	5
2.1 Introducere în Orchestrarea ETSI MANO/NFV	6
2.2 Arhitectura SONATA.....	7
2.3 Cadrul SONATA mapat la cadrul NFV ETSI	8
2.4 Experimente NFV realizate cu SONATA	8
Capitolul 3.....	9
Cadrul de Orchestrare OSM.....	9
3.1 Lucrări corelate -Prezentarea generală la nivel înalt OSM.....	9
3.2 Arhitectura OSM	10
3.3 Experimente NFV folosind OSM.....	12
Capitolul 4.....	12
Primitive și charm-uri ale Funcțiilor de Rețea Virtuale	12
4.1 Arhitectura	13
4.2 Juju, Primitve VNF și Charm-uri	14
4.3 Exemplu de proxy charm în VNF.....	14
Capitolul 5.....	16
Testarea și Monitorizarea QoS într-un model propus de arhitectură a serviciilor de network slicing.....	16
5.1 Concepte și studii de caz în 5G Network Slicing	17
5.2 Perspectiva asigurării calității serviciilor și provocările sale în Slicingul rețelei... 18	18
5.3 Contribuții pentru testarea și monitorizarea QoS într-un model de servicii de network slicing (tranșare a rețelei).....	18
5.3.1 Arhitectura propusă	19
5.3.2 Platforme folosite pentru arhitectura propusă.....	21
Capitolul 6.....	23
Concluzii	23
6.1 Rezultatele obținute	23
6.2 Contribuții originale	24
6.3 Lista lucrărilor originale	26
6.4 Perspective de dezvoltare ulterioară	30

Capitolul 1

Introducere

Cloud-ul oferă șansa de a transforma tehnologia într-un utilitar disponibil la nivel global, un activ dinamic care poate răspunde automat la creșterea cererii. Odată ce serviciile Cloud au evoluat de fapt într-o marfă de utilitate, fiecare companie trebuie să analizeze acum impactul Cloud și cum ar putea să obțină un avantaj competitiv.

Există o cerere a aplicațiilor actuale pentru a optimiza traficul managerial, virtualizarea resurselor comune, livrarea rapidă, în timp real a rețelei. Această nevoie a indus către conceptualizarea și adoptarea de noi modele de rețea, de exemplu Software Defined Network (SDN) [1][2] și Network Functions Virtualization (NFV)[3][4].

NFV promovează o decuplare între software și hardware-ul de bază prin implementarea diferitelor operațiuni de rețea, prin aplicații.

Explorările actuale în rețea au început să se concentreze asupra utilizării maxime atât a tehnologiilor avansate SDN, cât și a NFV și a avantajelor acestora.

Cu toate acestea, datorită complexității legate de conexiunea lor și de varietatea părților de rețea interconectate, asocierea lor poate prezenta anumite dificultăți. Standardizarea lor este în prezent cercetată, ceea ce complică executarea cadrului și principiilor lor complete.[1]

Tehnologiile de comunicații mobile 5G sunt apreciate ca o tehnologie care schimbă jocul, care poate susține o societate complet mobilă și conectată, acționând ca un ecosistem complet [5][6].

Una dintre capabilitățile puternice a tehnologiei 5G este aceea de feliere/sectionare a rețelei. Un grup controlat de subseturi de resurse, funcții de rețea / funcții de date virtuale de rețea în orice loc din lume, control, gestionare / orchestrare și planuri de servicii alcătuiesc un–Network Slice (NSL). O felie este o rețea / cloud logică end-to-end care funcționează pe o infrastructură partajată subadiacentă (fizică sau virtuală).

Feliile sunt separate unele de altele și pot fi manipulate și gestionate separat. Feliile pot fi comandate sau făcute la comandă.

Aplicațiile de rețea 5G sunt în principal comunicare ultra-fiabilă și cu latență redusă (URLLC), bandă largă mobilă îmbunătățită (eMBB) și comunicație de tip mașină (MTC) masivă și [7] și sunt implementate ca felii speciale.

1.1 Prezentarea domeniului tezei de doctorat

Domeniul principal al acestei teze este reprezentat de concentrarea pe tehnologia NFV și, în plus, studiul și contribuțiile proprii la două cadre NFV diferite: „SONATA (Switchless Optical Network for Advanced Transport Architecture)” 5Gtango și Open Source MANO (OSM) framework care urmează să fie studiate și dezvoltate în următoarele capitole.

Pe baza acestor cadre, domeniul tezei se concentrează în principal pe „Management și orchestrare (MANO) ; Managementul și integrarea ciclului de viață (LCM)” și asigurarea calității stivelor de tranșare a rețelei pentru NFV. Teza dezvoltă contribuții personale și experimente pentru VNF-uri, servicii de rețea (NS) și lanțuri de servicii, configurații software, concepte și instrumente.

În afară de aceasta, instrumentele și sistemele de cloud computing, SDN și modelare software în cloud sunt, de asemenea, incluse în studiu.

1.2 Scopul tezei de doctorat

Inițiativa acestei teze a fost inițial pornită din experiența mea profesională, însoțită de interesul pentru cercetarea științifică în NFV, virtualizare și software în cloud computing.

Mai mult, oportunitatea de a participa la un hackfest OSM [2019] și de a deveni membru dezvoltator în comunitatea lor mi-a adus șansa de a lucra și de a contribui la remedierea erorilor și la noi funcționalități cu privire la anumite structuri de cod (charm-uri) construite și executate în cadrul unei funcții de rețea virtualizată-VNF.

1.3 Conținutul tezei de doctorat

Această teză este organizată după cum urmează:

Capitolul 2 „Cadrul MANO-SONATA” prezintă cadrul SONATA și arhitectura acestuia și experimentele NFV folosind acest cadru.

Capitolul 3 „Cadrul MANO-OSM” prezintă arhitectura OSM și Openstack și principalele blocuri utilizate pentru contribuția personală la NFV, cu ajutorul experimentele făcute în participarea organizată Hackfest OSM Barcelona 2019

Capitolul 4 „Primitive si charm-uri ale funcțiilor de rețea virtuală” se concentrează pe înțelegerea charm-urilor (și primitivelor și rolurile lor în construcția VNF.

Capitolul 5 „Contribuții la asigurarea calității serviciilor în rețele slicing multi-domeniu prin VNF și charm-uri cu OSM” se concentrează pe soluții proprii în scopul testării și monitorizării calității serviciilor de rețea într-un scenariu de tranșare a rețelei.

Capitolul 6 „Concluzii” - rezumă principalele lecții învățate în cadrul activității tezei de doctorat și posibilele evoluții viitoare.

Capitolul 2

Managementul Orchestrării în cadrele de lucru NFV ETSI și SONATA

Acest capitol este dedicat înțelegerii conceptului de orchestrație în NFV. Ca punct de plecare, pentru a realiza arhitectura propusă în capitolul 5, am început cu analiza cadrelor ETSI, SONATA și OSM.

Ele reprezintă o implementare de bază majoră (din punct de vedere al orchestrării) în propunerea scenariului de monitorizare QoS a tezei din capitolul 5 (a se vedea figura 5.9), dar și pentru restul contribuțiilor personale din celelalte capitole. Prin urmare, a fost necesară introducerea celor mai relevante părți care au servit la teză.

2.1 Introducere în Orchestrarea ETSI MANO/NFV

În sprijinul administrării și orchestrării NFV, ETSI a produs un cadru arhitectural de referință și specificații [8]. Cadrul este conceput pentru a ajuta VNF-urile să ruleze pe mai mulți hipervizori și resurse de calcul.

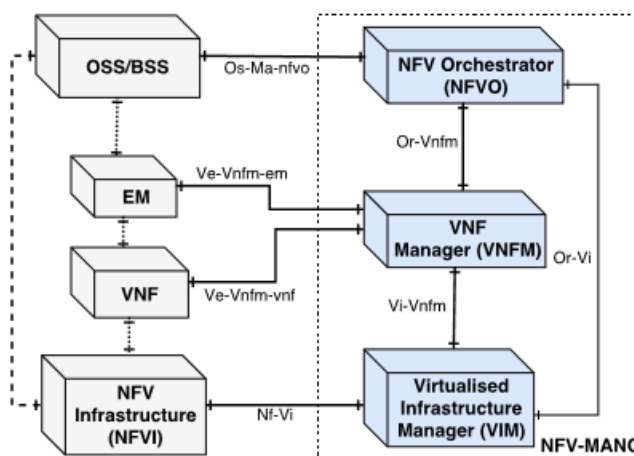


Fig.2. 1 Cadrul de lucru arhitectural ETSI-MANO [9]

Sistem de suport pentru Operare și Business (OSS / BSS): gestionează furnizarea și funcționarea serviciilor de rețea ale furnizorilor de servicii.

Managementul elementelor (EM): componentă responsabilă cu funcțiile tradiționale de gestionare a rețelei FCAPS (Defecțiuni, configurație, contabilitate, performanță și securitate) ale unui VNF care rulează.

VNF: este partea operațională executată pe un server fizic și reprezintă funcția de rețea de virtualizare. De exemplu, un firewall, un comutator, un router etc. sunt exemple de VNF-uri. Fiecare VNF individual este asociat cu un EM.

Infrastructura NFV (NFVI): reprezintă toate componentele hardware (calcul, stocare și rețea) și software în care sunt implementate, gestionate și executate VNF-urile.

MANO

Orchestratorul de virtualizare a funcțiilor de rețea (NFVO): este componenta principală, responsabilă de orchestrarea resurselor NFVI pe mai mulți manageri de infrastructură virtuală (VIM) și cel care realizează managementul ciclurilor de viață ale serviciilor de rețea. ”

Managerul VNF-urilor (VNFM): realizează configurarea și gestionarea ciclului de viață VNF (de exemplu, instanțierea, actualizarea, interogarea, scalarea, încetarea,

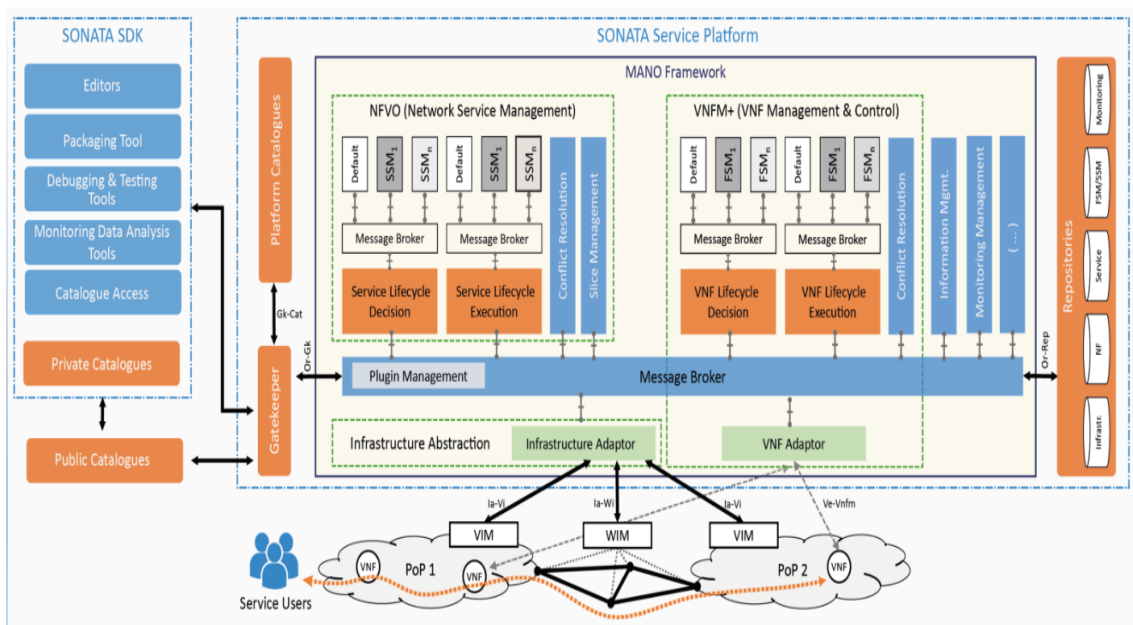
migrarea) pe domeniul său.” Mai mulți manageri VNF ar putea exista într-un anumit sistem.

VIM: bloc care asigură controlul și gestionarea resurselor NFVI, precum și interacțiunea unui VNF cu resursele hardware. De exemplu, OpenStack este o platformă cloud și OpenDaylight și Open Network Operating System (ONOS) sunt controlere SDN.

2.2 Arhitectura SONATA

Cadrul SONATA este construit pe baza paradigmei ETSI NFV, dar include instrumente suplimentare pentru a ajuta dezvoltorii să reducă durata pentru a obține un serviciu în rețea pe piață și, de asemenea să reducă și costurile serviciilor de rețea (NS).

După cum se poate observa în Fig.2.3, arhitectura generală a SONATA framework conține următoarele componente: Kit-ul de dezvoltare software-Software Development Kit (SDK), Platforma de servicii-Service Platform (SP) și cataloage în care se pot găsi diferite artefacte de sistem.



SDK-Service Development Kit; SP- Service Platform; VIM- Virtual Infrastructure Management

Fig.2.2 SONATA Framework [10]

Sursa pentru următoarele paragrafe (2.2 și 2.3) este preluată de la [https://sonata-nfv.eu/content/sonata-project] pentru a nu reduce sfera și sensul acesteia

SDK îi ajută pe dezvoltatorii terți să creeze servicii complexe compuse din mai multe VNF-uri, cu un set de instrumente software și susține, de asemenea, furnizorii de servicii pentru a implementa și gestiona serviciile de rețea create pe mai multe platforme de servicii .

Platforma de servicii (SP) este responsabilă de gestionarea și controlul funcțiilor și serviciilor de rețea. Este un mediu modular și personalizabil în care operatorii platformei pot crea platforme specifice adecvate modelului lor de afaceri, prin înlocuirea componentelor pluginurilor MANO.

Catалоgele constau în funcții de rețea și informații despre servicii precum cod, executabile, date de configurare și alte cerințe. Aceste catalоage sunt împărțite în platforme private, platforme de servicii și catalоage publice .

2.3 Cadrul SONATA mapat la cadrul NFV ETSI

Modului VIM gestionează resursele virtualizate (rețea, stocare și calcul) în domeniul infrastructurii unui operator atât în SONATA, cât și în ETSI (NFVI-PoP). Un VIM poate gestiona unul sau mai multe tipuri de resurse NFVI.

În comparație cu arhitectura ETSI, SONATA are o „componentă gatekeeper” care este responsabilă de evaluarea serviciilor de rețea transmise în platforma de servicii sub formă de pachete.

Drept urmare, SONATA adaugă extensii și contribuții pentru a îmbunătăți și a accelera asistența pentru chirie multiplă de resurse, precum și pentru a permite și autoriza felierea resurselor. (vezi Fig.2.4)

În comparație cu modelul ETSI, SONATA introduce SDK ca o nouă componentă arhitecturală. (simularea încercărilor de service, depanare și monitorizare și sprijinirea operațiunilor de serviciu de rețea DevOps.)

2.4 Experimente NFV realizate cu SONATA

Având în vedere complexitatea cadrului SONATA, aceste experimente și contribuția proprie au ca obiectiv principal să înțeleagă și să dovedească operabilitatea diferitelor VNF-uri în diferite topologii folosind cadrul SONATA..

Experimentele sunt:

- **Firewall virtual** - Obiective principale (legate de funcționalitatea de bază): crearea unui firewall virtual care are scopul de a bloca traficul între două gazde
- **Experiment grafic rutere virtuale**- Obiective principale: crearea unei mici rețele de routere virtuale care va redirecționa traficul printr-un grafic de rețea între trei gazde din trei subrețele diferite.

- Un alt experiment este crearea unei topologii de rețea al cărei scop este crearea mai multor VNF-uri cu platforma SONATA. Aceste roluri VNF sunt: gazde, routere, firewall, proxy, server http - toate virtuale

Testele de mai sus au folosit cadrul SONATA SDK complex pentru a demonstra adevărata funcționalitate și versatilitate a cadrului. Unele situații simple au fost proiectate în mod intenționat cu scopul de a evidenția și testa principalele operabilități ale instrumentelor.

Capitolul 3

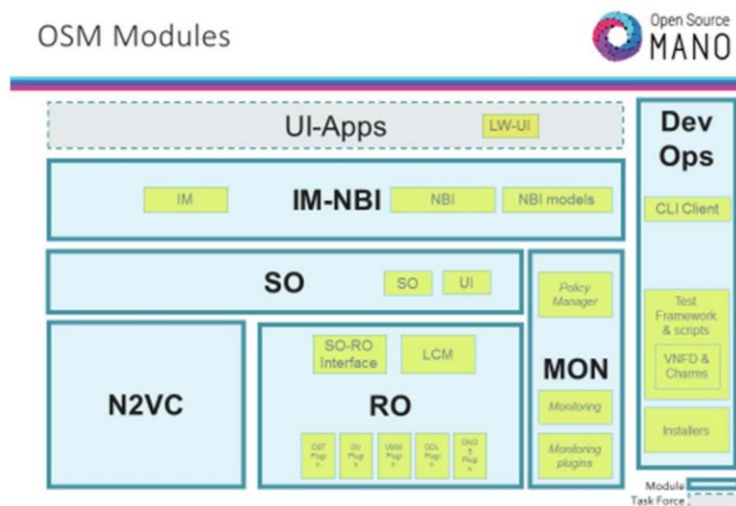
Cadrul de Orchestrare OSM

În acest capitol, a fost examinat și exploatat un alt cadru de orchestrație, adică Open Source MANO (OSM). Argumentul este că OSM integrează și conține cadrul SONATA anterior ca proiect, dar și ca funcționalități.

- Obiectivele generale ale acestui capitol și ale grupului de experimente sunt:
- Descrierea arhitecturii OSM, înțelegerea și instalarea cadrului OSM
- Înțelegerea și Instalarea Openstack
- Dezvoltarea serviciilor de rețea cu platforma OSM și Openstack
- Examinarea mai atentă și în amănunt a structurii descriptorilor în ceea ce înseamnă NFV și exemple și experimente VNF și NS.

3.1 Lucrări corelate -Prezentarea generală la nivel înalt OSM

Din punct de vedere al arhitecturii, OSM este în mod clar unul modular (Fig.3.1) și a progresat dobândind treptat principii de proiectare native pentru cloud. O caracteristică importantă pentru eficiența OSM și dezvoltarea comunității sale este modularitatea adaptivă a noilor funcționalități.



„*N2VC*- Serviciul de rețea pentru comunicarea cu VNF”; „*RO*- Orchestratorul de resurse”; „*MON*- Monitorizare”; „*Orchestratorul de servicii (SO)* e responsabil de orchestrarea și provizionarea serviciului până la capăt”; „*IM-NBI* – model informațional și interfața de nord”;

„**Fig.3. 1** Arhitectura OSM Generația a 5-a [12]”

IM-NBI reprezintă modelul informațional și interfața către nord și oferă automatizarea întregului ciclu de viață pentru funcții de rețea, servicii de rețea și felii de rețea.

Orchestratorul de servicii (SO) este folosit pentru a orchestra și livra un serviciu de la început până la sfârșit. În *SO*, sunt rezervate descrierea VNF și cataloagele NS.

Modulul Orchestratorul de resurse (RO) poate distribui servicii de rețea pe mai multe VIM-uri (VMware, OpenStack, OpenVIM și așa mai departe). De la ETSI MANO, aceste două componente, *SO* și *RO*, pot fi identificate ca NFVO. [13]

Modulul Serviciu de rețea către comunicația VNF (*N2VC*) este responsabil de cadrul de plugin al stratului de configurare și abstractizare VNF (*VCA*).

Stratul de Configurare și Abstractizare (VCA) este însărcinat să permită VNF-urilor și / sau Managerilor de elemente să trimită și să primească configurații, acțiuni și notificări.

3.2 Arhitectura OSM

Deoarece există multe detalii despre cadrul OSM și acestea sunt prezentate în teza principală, la această subsecțiune, rezumatul va conține imagini și informații

despre versiunea a cincea OSM și modelul de informații VNF și NS.

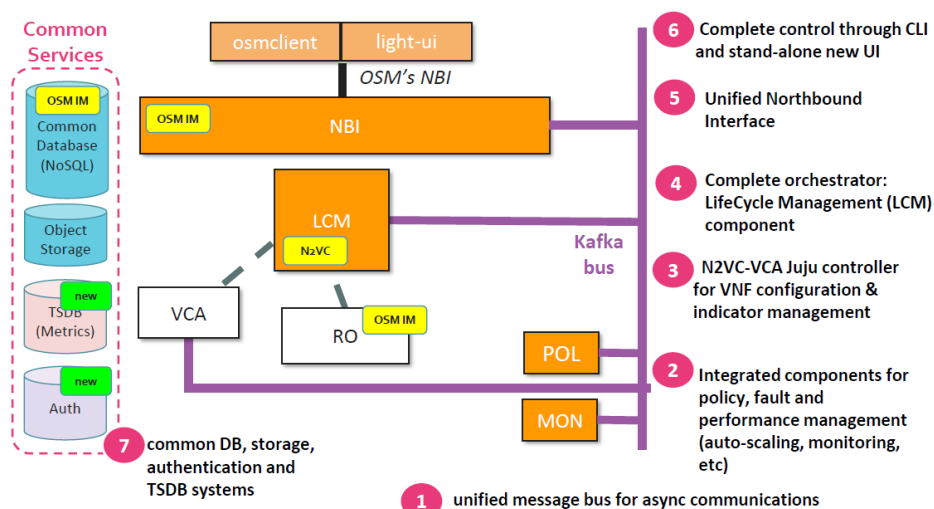


Fig.3.2 Arhitectura OSM- Generația a 5-a[14]

Datorită modularității și flexibilității sale, OSM își propune să reducă nevoile de integrare. Un model de informații bine cunoscut (IM) aliniat cu ETSI NFV este capabil să modeleze și să automatizeze întregul ciclu de viață al funcțiilor de rețea:

- VNFD (Descriptorul funcții virtuale de rețea) și VNFR (Virtual Network Function Record)

Un VNF poate fi compus din mai multe „Unități de implementare virtuală (VDU)”. „VNFD conectează VDU-uri utilizând legături virtuale interne (VL-uri)”. Fiecare VDU reprezintă un VM / container”. Fig. 3.4 ilustrează un exemplu de structură internă VNFD:

VDU- Virtual Descriptor Unit; *VLD*- Virtual Link Descriptor; *CP*- Connection point

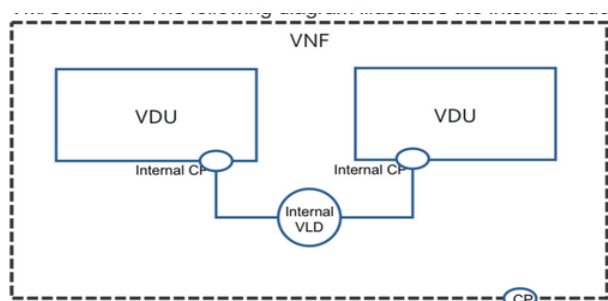


Fig.3.3 „Structura unui VNF reprezentată prin VDU- exemplu” [14]

Descriptorul de servicii de rețea [NSD] specifică o colecție de VNF interconectate care pot fi utilizate pentru a crea un serviciu de rețea cu numeroase VNF-uri. În plus, așa cum se vede în fig.3.5, NSD definește suplimentar informațiile de configurare la nivel NS.

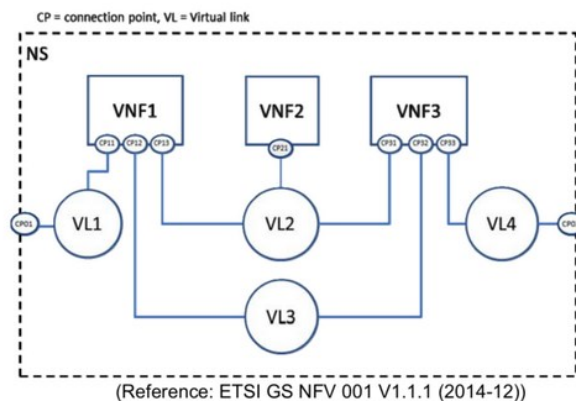


Fig.3.4 Descriptorul serviciului de rețea -exemplu[14]

3.3 Experimente NFV folosind OSM

Această secțiune prezintă experimentele personale NFV , al căror scop este de a crea diferite scenarii în OSM pentru evidențierea funcționalității diferitelor VNF-uri și NS-uri. Aceste experimente au fost realizate la al cincelea OSM Hackfest, Barcelona 4-8 februarie 2019 (un raport a fost oferit și universității, a se vedea Anexa 7). **Contribuția este personală, dar serverele și VIM-urile au fost oferite de comunitatea OSM în timpul hackfest-ului. Obiectivele atinse au fost: înțelegerea în detaliu a VNF-urilor și NS-urilor, cu ajutorul descriptorilor ; crearea de pachete VNF și NS.**

Capitolul 4

Primitive și charm-uri ale Funcțiilor de Rețea Virtuale

Această secțiune continuă lucrările din Capitolul 3, cu o secțiune despre cunoașterea și tratarea VNF-urilor, precum și ale instanțelor de charm-uri proxy , continuând cu un exemplu de eroare recurentă care apare în configurațiile de charm-

uri în OSM, prezentând soluția de rezolvare. Obiectivele generale ale acestui studiu și ale setului de experimente sunt:

- Însușirea noțiunilor legate de semnificațiile Ziua0- inițializare fundamentală; Ziua1-servicii de inițializare și Ziua2-operații executabile
- Analiza primitivelor și charm-urilor și cum sunt ele integrate
- Elaborarea și demonstrarea unui exemplu de proxy charm cu OSM

Principalele componente funcționale utilizare sunt: OSM, OpenStack, Juju, imagini pentru VIM.

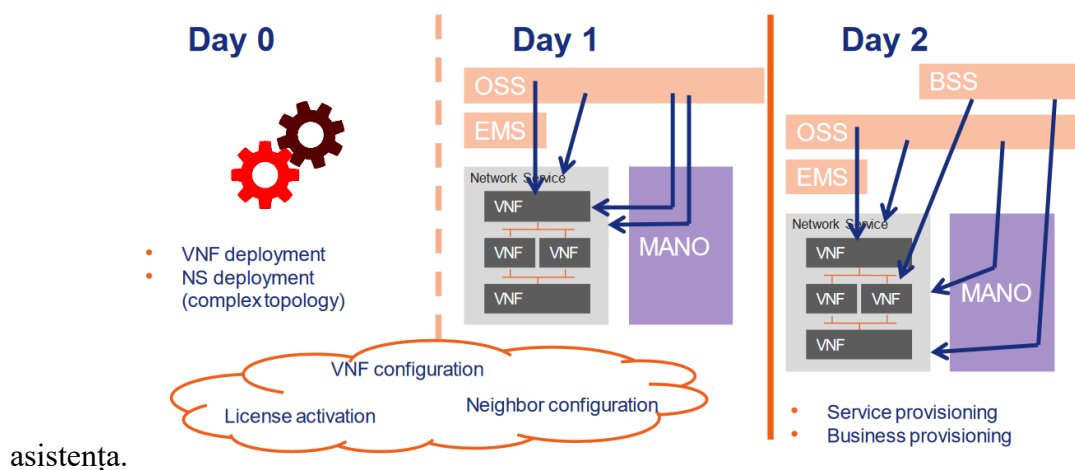
4.1 Arhitectura

Această subsecțiune este dedicată pentru „ziua 0, ziua 1, ziua 2,, construcții și configurații VNF. În același timp, se realizează introducerea primitivelor și charm-urilor VNF și, de asemenea, tranșarea în rețea”. Toate aceste componente fac parte din cadrul OSM. (vezi figura 4.1)

Ziua 0 este esențială pentru a avea grijă de planul de arhitectură și de resursele necesare (procesoare, spațiu de stocare, memorie RAM).

Software-ul care a fost elaborat în etapa „Ziua-0” este dezvoltat și implementat în „Ziua-1”. În pasul „Ziua-1”, configurația VNF trebuie să furnizeze serviciul programat pentru care a fost destinat.

În ziua 2, software-ul poate începe să crească în concordanță cu schimbările și comportamentul clientului. Pe lângă toate acestea, sunt introduse și întreținerea și



„Fig.4.1 Prezentare generală la nivel înalt Ziua 0, 1, 2 VNF [15]”

4.2 Juju, Primitve VNF și Charm-uri

Un charm[16] este un software care execută scripturi orientate asupra unui set de ținte (fie că vorbim de VNF, NS, aplicații șamd). Când charm-urile sunt create cu Juju, acestea sunt utilizate fie în cadrul unei aplicații, fie pe același sistem ca aplicația.

Juju [16] este un controler și un instrument gratuit specializat pentru operarea software-ului în cloud. În cadrul acestei teze, Juju este folosit în sensul elasticității modelului de operare a charm-urilor.

Charmul proxy este un concept care s-a născut în OSM pentru a avea instalat un software în VNF: charm-ul rulează în OSM, în containerul LXD și se conectează la aplicație cu modulul de la distanță. ” (vezi fig.4.3)

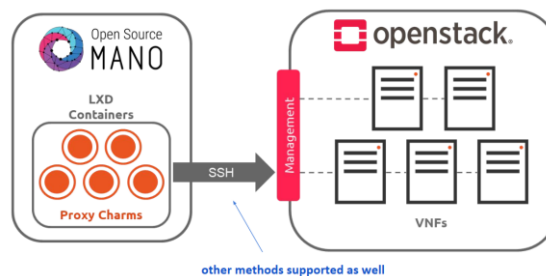


Fig.4.2 „Charm-uri Proxy în OSM” [16]

4.3 Exemplu de proxy charm în VNF

Această secțiune arată cum să structurați componentele necesare ale pachetului VNF pentru instanțierea corectă și configurarea de gestionare, astfel încât acestea să poată fi utilizate ulterior în proces.

Fiind membru al echipei de dezvoltatori OSM, am observat la un moment dat o eroare comună cu privire la implementarea charm-urilor. Acest lucru este legat de starea volumului lor de muncă. De-a lungul existenței lor, charm-urile pot trece prin diferite stări (cum ar fi starea de mentenanță, activă sau blocată), fie dacă este specificat din exterior sau nu. Unele probleme sau erori pot apărea în timpul procesului de implementare a charm-ului.

Problema obișnuită observată, bazată pe ceea ce a fost prezentat până acum, este legată de „accesul ssh din etapa Ziua 1”. Pentru acest moment particular, este important să existe doar „starea volumului de lucru activ” și să fie exclus restul stărilor de configurație care pot apărea aleatoriu.

Deci, un script python a fost dezvoltat odată cu contribuția mea, pentru asigurarea activă a stării sarcinii de lucru numai atunci când proxy-ul SSH este configurat corect. În figura 4.7 este script-ul implementat:

```
1 from charmhelpers.core.hookenv import (
2     action_get,
3     action_fail,
4     action_set,
5     status_set,
6 )
7 from charms.reactive import (
8     clear_flag,
9     set_flag,
10    when,
11    when_not,
12 )
13 import charms.sshproxy
14
15
16 @when('sshproxy.configured')
17 @when_not('simple.installed')
18 def install_simple_proxy_charm():
19     """Post-install actions.
20
21     This function will run when two conditions are met:
22     1. The 'sshproxy.configured' state is set
23     2. The 'simple.installed' state is not set
24
25     This ensures that the workload status is set to active only when the SSH
26     proxy is properly configured.
27     """
28     set_flag('simple.installed')
29     status_set('active', 'Ready!')
30
31
32 @when('actions.touch')
33 def touch():
34     err = ''
35     try:
36         filename = action_get('filename')
37         cmd = ['touch {}'.format(filename)]
38         result, err = charms.sshproxy._run(cmd)
39     except:
40         action_fail('command failed: ' + err)
41     else:
42         action_set({'output': result})
43     finally:
44         clear_flag('actions.touch')
```

Fig.4.3 Script implementat pentru corectare charm [17]

Capitolul 5

Testarea și Monitorizarea QoS într-un model propus de arhitectură a serviciilor de network slicing

Acest capitol prezintă principalele contribuții științifice bazate pe modul de testare și monitorizare a calității serviciilor (QoS) de rețea sau a lanțurilor de servicii în felii de rețea multi-domeniu. Se face o arhitectură propusă bazată pe OSM și o altă platformă de depanare. Simulările software sunt realizate cu charm-uri și primitive și lucruri care au fost deja prezentate în capitolele anterioare.

- Obiectivele generale ale acestui capitol sunt:
- Prezentarea tranșării rețelei în probleme generale și specifice în utilizarea cadrului OSM
- Analizarea provocărilor perspectivei QoS în tranșarea rețelei
- Propunerea unui scenariu științific, precum și o soluție practică pentru asigurarea QoS.
- Componentele funcționale și instrumentele utilizate sunt următoarele:
- OSM, Openstack, VmWare, Juju
- Soluția de asigurare „Infrastructură ca serviciu / platformă ca serviciu (IaaS / PaaS)”
- Charms –software scris în Python

5.1 Concepte și studii de caz în 5G Network Slicing

Slicingul de rețea este un termen folosit pentru a descrie cum se pot împărți și crea mai multe rețele virtualizate și logice pe infrastructuri multi-domeniu. O porțiune de rețea poate include anumite servicii și aplicații. SDN și NFV sunt alese pentru a lucra împreună pentru a conduce orchestrarea, administrarea și controlul, precum și instrumentele de analiză și automatizare.

Distribuirea în rețea poate furniza resurse radio, cloud și de rețea furnizorilor de aplicații și segmentelor verticale care nu au infrastructură fizică de rețea. Ca urmare, diferențierea serviciilor este activată prin adaptarea operațiunilor de rețea pentru a se potrivi nevoilor clienților în funcție de tipul de serviciu. [18]

Principalele considerații pe care OSM le-a implementat din cadrul ETSI NFV, în ceea ce privește tranșarea rețelei sunt:

- a. posibilitatea de a face compoziția serviciului de rețea într-o felie. Aceasta înseamnă o modalitate de a alocă unui vecin o felie din mai multe servicii de rețea și că aceste servicii de rețea vor fi conectate în interiorul feliei respective.
- b. Al doilea lucru important este conceptul de partajare a NST (Network Slice Templates). Această partajare are aspirații pentru realizarea resurselor și conectarea diferitelor felii.

În ceea ce privește implementarea, iată pașii principali:

- fiecare felie este construită ca un grup de servicii de rețea conectate prin rețele sau descriptori de legături virtuale (VLD)
- clasificarea feliei construite la cerere este determinată cu parametri simpli de intrare
- pentru a îndeplini cerințele NST, sunt necesare VNFD și NSD (detaliile pot fi găsite la 5.3.1)
- înainte de instanțierea slice-urilor, pachetele VNF și NS trebuie adăugate dezvoltării unui NSI

5.2 Perspectiva asigurării calității serviciilor și provocările sale în Slicingul rețelei

Această secțiune este menită să introducă concepte generale despre „Calitatea serviciului (QoS) și Calitatea experienței (QoE) în tranșarea rețelei 5G” și, de asemenea, provocări care pot apărea în implementarea lor pentru satisfacția clienților. Este o scurtă descriere proprie făcută din cercetarea lucrărilor reale pe acest subiect, precum și din proiecte specifice: SliceNet, SONATA / 5GTANGO, 5GNORMA și OSM.

Rețelele 5G trebuie să aibă următoarele capacități QoS [19] pentru a satisface cerințele QoS complicate ale diferitelor aplicații și servicii:

- Evenimentele legate de QoS din trecut pot fi reconstituite automat și cu precizie;
- Evenimentele actuale legate de QoS pot fi identificate cu precizie pentru a declanșa atenuarea automată sau acțiunile imediate propuse. Evenimentele viitoare legate de QoS pot fi prezise cu precizie

5.3 Contribuții pentru testarea și monitorizarea QoS într-un model de servicii de network slicing (tranșare a rețelei)

În această secțiune, se face o scurtă analiză și comparare a metodelor de monitorizare active și pasive într-o rețea.

Pe baza acestui fapt, ideea principală a acestei teze este de a arăta un exemplu de testare și monitorizare în timp real pentru KPI-uri specifice dintr-un scenariu de feliere a modelului de serviciu.

Mai mult, exemplul sugerat demonstrează modul în care testarea NS poate fi automatizată în medii live. După cum va fi prezentat în continuare, monitorizarea și testarea codului virtual pot fi realizate și introduse în tranșarea rețelei prin intermediul charm-urilor care sunt orchestrate de OSM MANO pentru a captura KPI-uri specifice pentru diferite lanțuri de servicii, cu ajutorul factorilor/agenților de charm virtual. (vCF) Se numește virtual deoarece poate fi configurat în / ca / sau atașat la un VNF / NS și este cod software prin charm.

5.3.1 Arhitectura propusă

Modurile clasice de până acum în ceea ce privește durata de viață continuă a supravegherii serviciilor sunt de a prelua date din infrastructură, de la routere, comutatoare etc. și de a le transfera într-un plan de date mari, dar nu se ajunge la tot drumul. Un serviciu poate fi testat și monitorizat cap la cap dintr-un lanț de servicii folosind arhitectura sugerată, așa cum se arată în Figura 5.9. Scopul este de a arăta cum testarea și monitorizarea serviciilor de rețea (NS) poate fi organizată și automatizată în contexte dinamice.

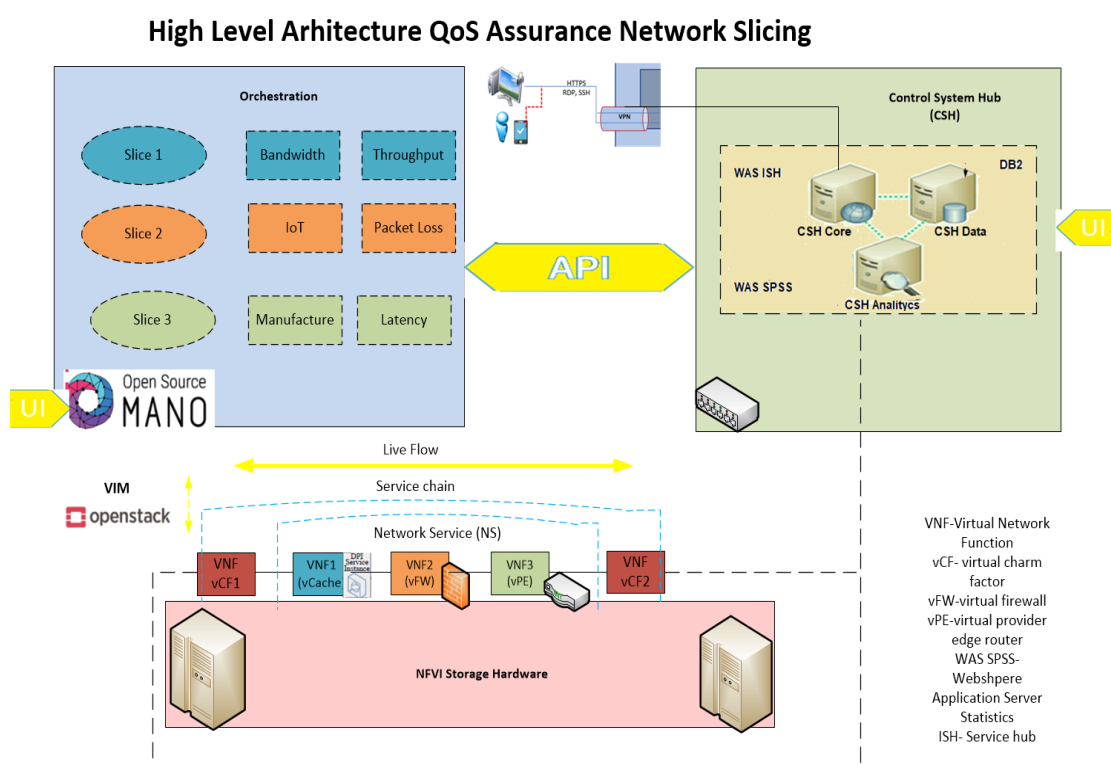


Fig.5.1 Arhitectură QoS 5G Network Slicing

Această arhitectură este propunerea mea

Dintr-o perspectivă la nivel înalt, arhitectura scenariului de tranșare propus constă din următoarele părți:

- - OSM face orchestrarea, va implementa VNF, NS, lanțul de servicii și felii, inclusiv agenții de testare.

- *Control System Hub (CSH)* este platforma responsabilă cu coordonarea vCF-urilor și interacționează cu orchestratorii prin API. De asemenea, oferă fluxuri de lucru de automatizare
- *Openstack folosit ca VIM* creează interoperarea între componente

În scenariul de slicing există trei felii diferite pentru diferite tipuri de servicii:

- *Slice dedicat lățimii de bandă* este destinat abonaților rezidențiali. Randamentul este KPI pentru asigurarea SLA. Într-un scenariu real, un VCache VNF este potrivit pentru acesta.
- *IoT-* Cazul de utilizare a feliei dedicat IoT necesită pierderi reduse de pachete. Un firewall virtual (vFW) a fost ales ca fiind potrivit pentru VNF
- *Industrial-* Felia dedicată industriei dorește de obicei o latență scăzută.

Arhitectura nu se concentrează asupra modului în care sau de ce au fost alese aceste felii (acestea pot fi considerate experimentale, pe baza cererilor reale ale clienților în contextul 5G și al tranșării rețelei), „*ci cum să monitorizăm constant traficul în timpul utilizării normale a rețelei, astfel încât problemele legate de calitatea rețelei să poată fi urmărite și raportate dinamic*”. Acest lucru se realizează cu inovația vCF, care este practic un VNF activ și acționează ca un agent de testare virtuală; (vezi Figura 5.10):

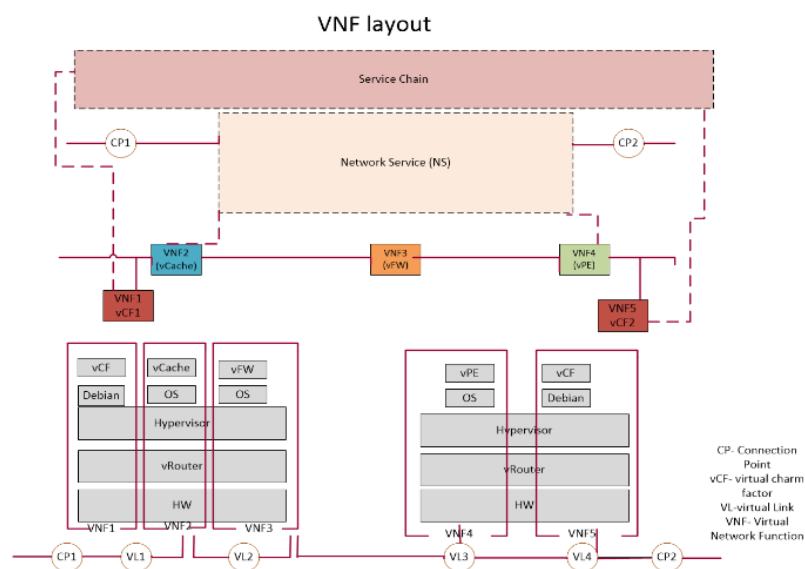


Fig.5.2 Lanțul de servicii și structura VNF-urilor

1. Design-ul și planul elementelor descriptorilor VNF- VNFDs (în OSM)
2. Design-ul și planul elementelor descriptorilor NS-NSD (în OSM)
3. Design-ul și activarea modelelor de testare și monitorizare (în CSH)

4. Instanțierea serviciilor de rețea și a network slice-urilor
5. Inițializarea VNF și a agenților de testare (vCF) și interconectarea lor în lanțul de servicii
6. Apelare CSH și alocarea lor (VNF, vCF) în inventar
7. Rulare declanșator din configurația inițială de primitive în scopul activării unui serviciu de testare
8. KPI preluat de OSM în timp real-Figura 5.15

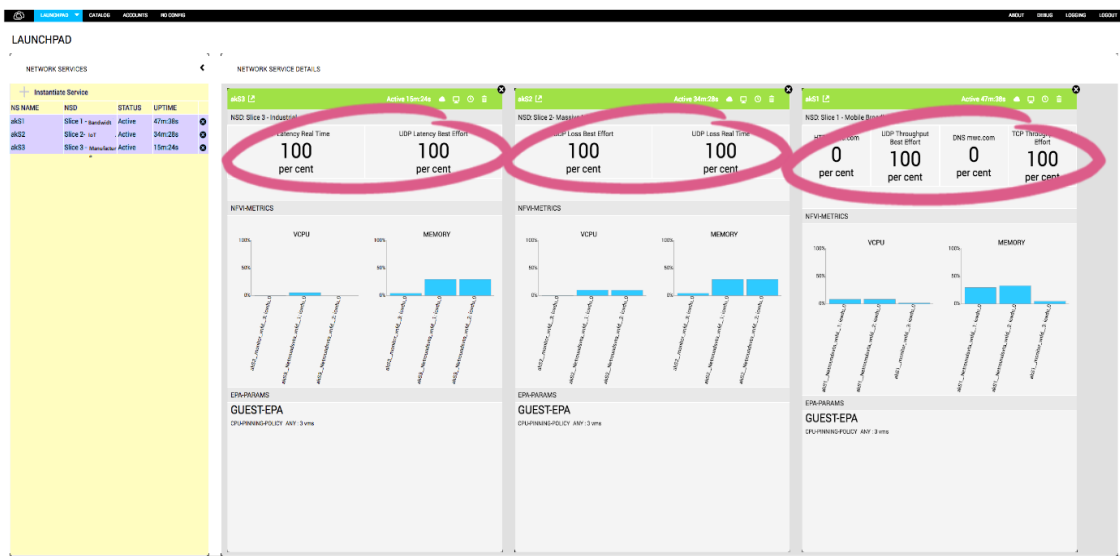


Fig.5.3 Real-Time SLAs

Feliile corespunzătoare sunt monitorizate 24/7, deci ceea ce se vede aici, aspectul verde este în timp real, feliile sunt monitorizate continuu și granularitatea este de 10 secunde.

CSH calculează erorile în secunde. - Dacă vreunul dintre pragurile specificate este încălcat, acestea vor fi detectate și reflectate înapoi la OSM, astfel încât OSM va acționa fie pentru redistribuire, fie pentru acțiunea în acea porțiune specifică.

5.3.2 Platforme folosite pentru arhitectura propusă

Pentru hardware-ul de **stocare NFVI**, au fost utilizate **servele de la IBM**. Pentru partea **VIM**, **Openstack** a fost folosit pentru a fi partea de cloud computing a arhitecturii. O altă parte importantă de la VIM este rețeaua. Rețeaua fizică constă din comutator fizic configurat și, de asemenea, un dispozitiv de firewall de poartă - Vyatta Brocade, care acționează ca un router, precum și ca un firewall, pentru a prelua controlul asupra întregului trafic fizic din interiorul centrului de date și între ambele. centre de date (OSM și CSH). Pentru rețeaua virtuală, a fost aleasă funcția de comutare standard în **VMware vCenter**. Aceasta este o rețea de mașini virtuale care

rulează pe un singur sistem fizic și sunt conectate logic astfel încât să poată transmite și primi date unul de la altul.

În afară de aceasta, serverele virtuale Domain Name System (DNS) și Domain Controller au fost configurate și în VIM în interiorul unei mașini virtuale (VM) cu Windows Server 2018 ISO. Aceste roluri sunt necesare pentru ca toate VM-urile, VNF-urile și VCF-urile să poată comunica între ele.

Pentru partea **OSM**, pe lângă proiectarea și configurarea VNFD-urilor, NSD-urilor și a feliilor de rețea (prezentate deja în 5.3.1), o altă parte importantă de care este responsabil OSM, este **instanțierea**.

Platforma **CSH** conține în mod tipic trei servere: componentele Data, Core și Analytics. La partea de bază, componenta principală constă din **IBM® WebSphere® Application Server ISH** (Service hub) care servește aplicații de la front office la back office. **CSH Analytics** este locul în care este generat **traficul pentru vCF-uri**. Dispune de software client IBM SPSS Analytics pentru explorarea, raportarea și modelarea detaliată a datelor.

În CSH Data există informații stocate despre șabloanele vCF-urilor și, de asemenea, stochează rapoartele de monitorizare care pot fi trimise clienților.

După ce fiecare felie este instanțiată și trece prin testul de activare, OSM cere CSH să instaureze monitorizarea continuă a serviciului live (a se vedea Figura 5.21). VCF-urile declanșează traficul către lanțul de servicii pentru a certifica SLA pe termen real. Pentru a captura KPI-urile, o caracteristică cheie este introducerea vCF-uri, dar și în diferite locații din calea traficului.

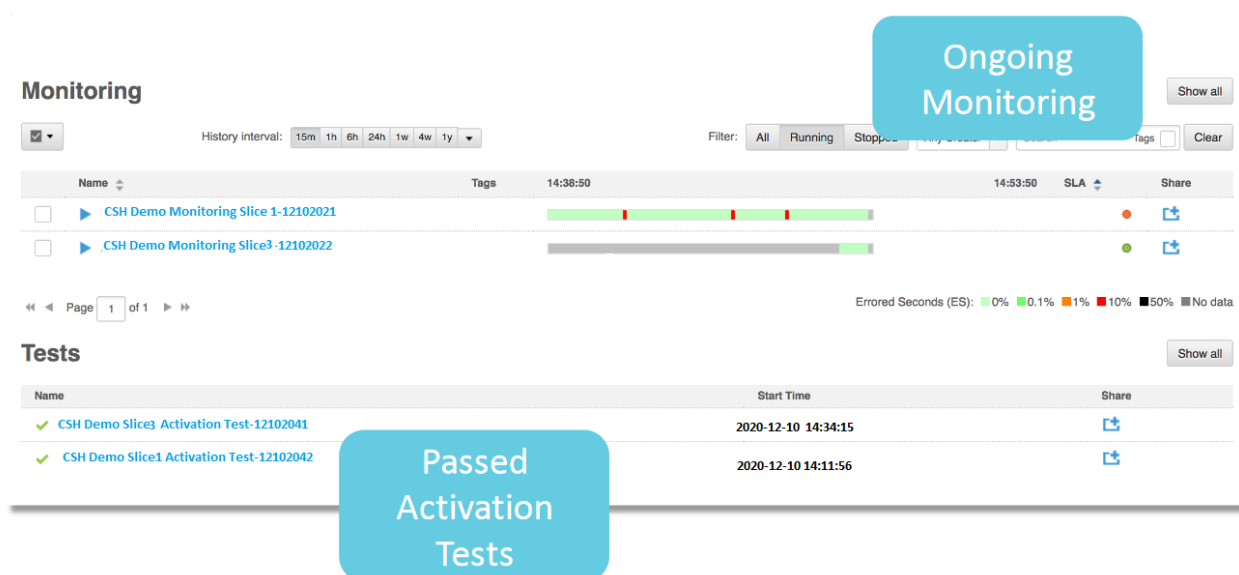


Fig.5.4 Testare și Monitorizare în CSH

Capitolul 6

Concluzii

Abordarea NFV se concentrează în general pe managementul și sănătatea VNF din punct de vedere al resurselor, această teză intenționată prezintă necesitatea unui test de calitate al NS general.

Monitorizarea activă este un răspuns adecvat pentru acest lucru, cu integrarea vCF-urilor. Aceștia pot detecta cazuri anterioare legate de modul în care se pot obține informații despre asigurarea calității din viziunea utilizatorilor finali și pot aprecia ce părți ale rețelei sunt afectate.

Testele și demonstrative pot fi efectuate pe baza regulilor firewall-ului, a latenței, a transferului, a priorității QoS etc. Agenții de monitorizare vCF au capacitatea și versabilitatea de a fi poziționați la orice nivel în cadrul design-ului unui VNF și/sau serviciu de rețea NS: fie la nivelele de bază ale sistemelor de operare OS, la nivelul hypervizorilor, în cadrul descriptorilor și primitivelor de configurare sau de-a lungul întregii căi a lanțului de serviciu final.

Teza de lucru actuală prezintă trei felii de rețea diferite cu setări QoS diferite și demonstrează modul în care monitorizarea activă și testarea contribuie la livrarea unei QoS felii potrivite pentru clienți.

6.1 Rezultatele obținute

Capitolul 1, Introducere prezintă teza, motivația și structura sa. Domeniul tezei se concentrează foarte mult pe management și orchestrare MANO, managementul ciclului de viață al VNF/NS - LCM și asigurarea calității stivelor de tranșare a rețelei pentru VNF, dezvoltând experimente personale și contribuția la VNF-uri, servicii de rețea (NS) și lanțuri de servicii prin configurații software, concepte și instrumente. Cadrele utilizate sunt SONATA 5Gtango și OSM.

În capitolul 2, ETSI MANO, arhitectura SONATA și aspectele lor relaționale au fost dezbătute. Au fost prezentate contribuțiile proprii la experimentele NFV folosind acest cadru (vezi 6.2)

Capitolul 3 a fost complet alocat pentru OSM: prezentare la nivel înalt, arhitectură și experimente făcute la hackfest OSM.

Rezultatele din capitolul 4 se bazează pe primitive și charm-uri VNF. Există, de asemenea, un exemplu de proxy charm cu o contribuție proprie de cod de remediere a erorilor. (vezi 6.2)

Capitolul 5 este în întregime opera autorului atât din abordări teoretice, cât și practice. Accentul principal se bazează pe network slicing și provocările care pot apărea la satisfacția clienților din perspectiva asigurării nivelului calității serviciilor. Datorită acestui fapt, au fost propuse un scenariu și o arhitectură de rețea QoS proprii, împreună cu contribuția proprie pentru platformele de testare și monitorizare pentru a ajuta atât furnizorii de servicii (SP-urile), cât și clienții să îndeplinească cerințele serviciului. (vezi 6.2)

Capitolul 6 este alocat pentru concluzii și perspective viitoare.

6.2 Contribuții originale

1. Paralelă între arhitectura cadrului SONATA și modelul ETSI NFV.
Contribuțiile pot fi găsite în: Capitolul 2.3 Publicații: [1-COMM18]
2. „Experimentul firewall-ului virtual VNF care are scopul de a bloca traficul între două gazde. Scopul este de a testa funcționalitatea diferitelor VNF-uri în diferite topologii utilizând cadrul SONATA. Aceste topologii sunt reprezentate ca rețele emulate personalizate care folosesc containere Docker ca instanțe de calcul pentru a rula VNF-uri. ”
Contribuțiile pot fi găsite în: Capitolul 2.4 Publicații: [2-SOFT18]
3. Experiment grafic rutere virtuale. Obiective principale: crearea unei mici rețele de rutere virtuale care va redirectiona traficul printr-un grafic de rețea între trei gazde din trei subrețele diferite.
Contribuțiile pot fi găsite în: Capitolul 2.4 Publicații: [2-SOFT18]

4. Un alt experiment este o topologie de rețea al cărei scop este instanțierea VNF-urilor cu platforma SONATA. Aceste roluri VNF sunt: gazde, routere, firewall, proxy, server http, toate virtuale. Testele au dovedit cu succes că accesul la serverul http prin serverul proxy a funcționat fără o rută cunoscută și, de asemenea, că firewall-ul a filtrat traficul de intrare către proxy prin blocarea unei anumite rețele.

Contribuțiile pot fi găsite în: Capitolul 2.4 Publicații: [1-COMM18]

5. Vizualizare comparativă între cadrul SONATA și modelul ETSI.

Contribuțiile pot fi găsite în: Capitolul 2.4 Publicații: [1-COMM18]

6. Lucrări conexe, scurtă paralelă între SONATA și OSM și un studiu al cazurilor reale de utilizare a OSM. Această parte a fost necesară pentru a integra OSM în infrastructură și experimente.

Contribuțiile pot fi găsite în: Capitolul 3.1 Publicații: [3-IARIA ICN20]

7. Experimente NFV al căror scop este de a testa funcționalitatea diferitelor VNF-uri în diferite topologii utilizând cadrul OSM. Aceste experimente au fost realizate la 5 OSM Hackfest, Barcelona 4-8 februarie 2019 (un raport a fost oferit și universității). Contribuția muncii este personală, dar serverele, VIM-urile și etapele fluxului de lucru au fost oferite de comunitatea OSM în timpul hackfestului.

Contribuțiile pot fi găsite în: Capitolul 3.3 Proiect raport tehnic: [A.7 OSM Hackfest19]

8. Vizualizare selectivă a primitivelor și charm-urilor VNF, toate integrate cu OSM. Contribuție proprie: dezvoltarea unui script python pentru a controla starea sarcinii de lucru să fie activată numai atunci când proxy-ul SSH este configurat corect.”

Contribuțiile pot fi găsite în: Capitolul 4.2 Publicații: [4-IARIA AICT20]

9. Dezbaterea provocărilor perspectivei QoS în network slicing printr-o scurtă analiză și comparație pentru QoS și QoE urmând să fie prezentate, pe baza unor proiecte specifice.

Contribuțiile pot fi găsite în: Capitolul 5.2 Publicații: [ScientificBuletin]

10. Arhitectură la nivel înalt QoS Assurance Network Slicing
Contribuțiile pot fi găsite în: Capitolul 5.3.1 Publicații: [ScientificBuletin]

11. Platforme utilizate pentru arhitectura propusă și configurația rețelei.

Contribuțiile pot fi găsite în: Capitolul 5.3.2 Publicații: [BlackSeeConf]

12. Cod software detaliat

Contribuțiile pot fi găsite în: Capitolul 5.3.3 și Anexa 6

6.3 Lista lucrărilor originale

Lista lucrărilor publicate sau acceptate spre publicare

1. „A. Țapu, C. Conțu, E. Borcoci, *Multiple Chained Virtual Network Functions Experiments with SONATA Emulator*, The Twelfth International Conference on Communications COMM 2018”

Published in: [2018 International Conference on Communications \(COMM\)](#)

Date of Conference: 14-16 June 2018

Date Added to IEEE Xplore: 08 October 2018

ISBN Information:

Electronic ISBN:978-1-5386-2350-3

USB ISBN:978-1-5386-2349-7

Print on Demand(PoD) ISBN:978-1-5386-2351-0

DOI: 10.1109/ICComm.2018.8484762

Publisher and Indexed in : IEEE BDI Conference

Conference Location: Bucharest, Romania

ISI Web of Science Number: WOS:000449526000053 ISBN:978-1-5386-2350-3, I SSN: 1550-3607

2. „A. Țapu, C. Conțu, E. Borcoci, *Network Function Virtualization Experiments using SONATA Framework*, The International Symposium on Advances in Software Defined Networking and Network Functions Virtualization SOFTNETWORKING 2018”

Published in: [ICN 2018](#), The Seventeenth International Conference on Networks

April 22, 2018 to April 26, 2018 - Athens, Greece ISBN: 978-1-61208-625-5

<https://www.iaria.org/conferences2018/ProgramICN18.html>

awarded paper

https://www.iaria.org/conferences2018/awardsICN18/icn2018_a5.pdf

Also published in Journal

https://www.iariajournals.org/networks_and_services/tocv11n34.html

3. C.Conțu, **A.Ciobanu** E.Borcoci, „*Virtual Network Function Use Cases Implemented on SONATA Framework*”, - extension of paper4 no.2, International Journal on Advances in Networks and Services published by IARIA.ISSN:1942-2644 journals site: <http://www.iariajournals.org> vol. 11, 2018, No. 3&4, pages: 103-112,
https://www.iariajournals.org/networks_and_services/netser_v11_n34_2018_paged.pdf
4. „**Andra Ciobanu**, Cosmin Contu, Eugen Borcoci "*Study on Use-Cases of Open Source Management and Orchestration Framework in 5G Projects*"- 2020 - The Nineteenth International Conference on Networks”
Published in: [ICN 2020](#), The Nineteenth International Conference on Networks
February 23, 2020 to February 27, 2020 - Lisbon, Portugal
ISBN: 978-1-61208-770-2
- awarded paper: <http://www.iaria.org/conferences2020/AwardsICN20.html>
http://www.thinkmind.org › icn_2020_2_30_38009
5. „**Andra Ciobanu**, Cosmin Contu, Eugen Borcoci "*Charms and Virtual Network Functions primitives Experiments using Open Source MANO framework*" - the Sixteenth Advanced International Conference on Telecommunication”
Published in: [AICT 2020](#), The Sixteenth Advanced International Conference on Telecommunications
<https://www.semanticscholar.org/paper/Charms-and-Virtual-Network-Functions-Primitives-Ciobanu-Borcoci/3580ef2158b5dbcfe054f4b1ea9b0515a23a3c0a>

http://www.thinkmind.org/index.php?view=article&articleid=aict_2020_1_20_10029

awarded paper <https://www.iaria.org/conferences2020/AwardsAICT20.html>

6. Eugen Borcoci, *Andra Ciobanu*, Cosmin Contu , *"Layered Network Domain Resource Management in Multi-domain 5G Slicing Environment"*, ADVANCED INFORMATION AND COMMUNICATION TECHNOLOGIES-2019

Published in: www.aict.info/2019 AICT 2019 : The Fifteenth Advanced International Conference on Telecommunications

https://www.thinkmind.org/articles/aict_2019_...

<https://www.semanticscholar.org/paper/Layered-Network-Domain-Resource-Management-in-5G-Borcoci-Ciobanu/0da977fb068d96c5bdc0b689b6b34e454abf6083>

7. E.Borcoci, C.Contu, *A.Ciobanu* „5G Slicing Management and Orchestration Architectures - Any Convergence?”, The Eleventh International Conference on Advances in Future Internet AFIN 2019, October 27, 2019 to October 31, 2019 - Nice, France, <https://www.iaria.org/conferences2019/AFIN19.html>

8. Eugen Borcoci, Cosmin Contu, *Andra Ciobanu* " *On Heterogeneity of Management and Orchestration Functional Architectures in 5G Slicing*, IARIA 2019"

Published in: International Journal on Advances in Internet Technology, vol 13 no 1 & 2, year 2020, http://www.iariajournals.org/internet_technology/

Paper Invited for IARIA Journals

<https://www.iaria.org/conferences2019/AwardsAICT19.html>

9. E.Borcoci, C.Contu, *A.Ciobanu*, *extension of the Paper no.8*, "On Heterogeneity of Management and Orchestration Functional Architectures in 5G Slicing", [International Journal On Advances in Internet Technology](http://www.iariajournals.org/internet_technology), vol 13, no 1&2, year 2020, http://www.iariajournals.org/internet_technology/tocv13n12.html

10. *Andra Ciobanu*, Eugen Borcoci, Marius Vochin „*SCENARIO FOR OPTIMUM SERVICES POSITION WITH CONTROL SYSTEM HUB AND*

OPEN SOURCE MANO IN EDGE COMPUTING", Scientific Buletin 2021

Accepted and In course of Publishing:

<http://www.scientificbulletin.upb.ro/openconf/data/fromreviewer/14505.pdf>

Submission Number 11125

Indexed in ISI and SCOPUS

11. Andra Ciobanu, Eugen Borcoci, Marius Vochin, „**A Quality-of-Service Scenario Awareness for Use-Cases of Open-Source Management and Control System Hub in Edge Computing**”, BlackSee Conference 2021

Published

in:

https://www.academia.edu/49040476/A_Quality_of_Service_Scenario_Awareness_for_Use_Cases_of_Open_Source_Management_and_Control_System_Hub_in_Edge_Computing

Indexed in: IEEE BDI Conference

12. Pantelimon-Teodor Tivig, Eugen Borcoci, Alexandru Brumaru, **Andra-Isabela-Elena Ciobanu**, „*Layer 3 Forwarder Application - Implementation Experiments Based on Ryu SDN Controller*”, IEEE ISNCC Dubai Conference 2021

Accepted, waiting for publishing: <https://www.isncc-conf.org/> International Symposium on Networks, Computers and Communications Dubai 2021

Indexed in: IEEE BDI

13. **Andra-Isabela-Elena Ciobanu**, Cosmin Contu, Eugen Borcoci, Marius-Constantin Vochin, Frank Y.Li, „*Optimal Service Placement with QoS Monitoring in NFV and Slicing Enabled 5G IoT Networks*”, Globecom GC 2021 Workshop IoST5G&B

Submitted July 2021 – waiting for acceptance : Indexed in: IEEE BDI

14. Eugen Borcoci, **Andra Tapu**, Cosmin Contu, “*D1.2 Virtual Evolved Packet Core*”, ORANGE- Romania – UPB cooperation project "5G Technologies", 2018

Rapoarte de cercetare științifică de doctorat

A.Ciobanu, „Virtualization methods for network,systems and servers in cloud”, Ph.D Scientific Research Report 1 (unpublished), University Politehnica of Bucharest, Romania, April 2017

A.Ciobanu, „Technologies of Cloud Computing used in heterogenous modern networking”, Ph.D Scientific Research Report 2 (unpublished), University Politehnica of Bucharest, Romania, December 2017

A.Ciobanu, „Network Services and Slices Experiments using OSM framework-Part 1”, Ph.D Scientific Research Report 3 (unpublished), University Politehnica of Bucharest, Romania, July 2018

A.Ciobanu, „Charms and Virtual Network Functions Primitives and network slices experiments using OSM framework-Part2”, Ph.D Scientific Research Report 4 (unpublished), University Politehnica of Bucharest, Romania, January 2019

Lista de Proiecte

Colaborare între UPB și Orange România SLICENET:End-to-End Cognitive Network Slicing and Slice Management Framework in Virtualised Multi-Domain, Multi-Tenant 5G Networks”, February-May 2018

„A Massive MIMO Enabled IoT Platform with Networking Slicing for Beyond 5G IoV/V2X and Maritime Services”, November 2020-în curs

6.4 Perspective de dezvoltare ulterioară

Ultimul proiect menționat mai sus este munca în curs și viitoare pe aceeași zonă de network slicing. Intenția este de a aduce contribuții până la sfârșitul acestui proiect și de a participa și la alte oportunități. Zona de interes va fi păstrată pe NFV, SDN, cloud computing, dar și pe tehnologiile tangențiale moderne și noi.

Din perspectiva academică, este luată în considerare și posibilitatea de a intra în ierarhia cadrelor didactice

Bibliografie

- [1] Qiang Duan, Nirwan Ansari, and Mehmet Toy, Software-defined network virtualization: An architectural framework for integrating SDN and NFV for service provisioning in future networks, IEEE Network, 2016
- [2] N. Omnes, M. Bouillon, G. Fromentoux, and O. le Grand, “A programmable and virtualized network & IT infrastructure for the internet of things: How can NFV & SDN help for facing the upcoming challenges,” in 2015 18th International Conference on Intelligence in Next Generation Networks, ICIN 2015, Mar. 2015, pp. 64–69, doi: 10.1109/ICIN.2015.7073808
- [3] ETSI GS NFV-SWA 001 V1.1.1, Network Functions Virtualisation (NFV); Virtual Network Functions Architecture, 2014
- [4] Manuel Paul, Sibylle Schaller, Malcolm Betts, Dave Hood, Meral Shirazipour, Diego Lopez, John Kaippallimalil, Applying SDN Architecture to 5G Slicing, USA, 2016
- [5] NFV White paper: “Network Functions Virtualisation, An Introduction, Benefits, Enablers, Challenges & Call for Action. Issue 1”. [Online]. Available from: https://portal.etsi.org/NFV/NFV_White_Paper.pdf
- [6] ETSI Industry Specification Group (ISG) NFV, 2013. ETSI GS NFV 002 V1.1.1: Network Functions Virtualisation (NFV); Architectural Framework. Technical Report.
URL: http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf
- [7] Anderson, J.L., Autonomous systems intelligence, in: Proceedings of the 1983 annual conference on Computers : Extending the human resource - ACM 83, ACM Press, New York, New York, USA, pp. 229–233, 1983 URL: <http://portal.acm.org/citation.cfm?doid=800173.809741>, doi:10.1145/800173.809741
- [8] A. Țapu, C. Conțu, E. Borcoci, “Multiple Chained Virtual Network Functions Experiments with SONATA Emulator”, The Twelfth International Conference on Communications COMM 2018.
- [9] ETSI Industry Specification Group (ISG) NFV, a. ETSI GS NFV-EVE 005 V1.1.1: Network Functions Virtualisation (NFV); Ecosystem; Report on SDN Usage in NFV Architectural Framework. Technical Report. ETSI. URL:

http://www.etsi.org/deliver/etsi_gs/NFV-EVE/001_099/005/01.01.01_60/gs_nfv-eve005v010101p.pdf.”

[10] Manuel Peuster, Stefan Schneider and contributors, SONATA. “D2.2 Architecture Design”, February, 2018, Available from: http://sonatanfv.eu/sites/default/files/sonata/public/contentfiles/pages/SONATA_D2.2_Architecture_and_Design.pdf.

[11] A. Țăpu, C. Conțu, E. Borcoci, “Network Function Virtualization Experiments using SONATA Framework”, The International Symposium on Advances in Software Defined Networking and Network Functions Virtualization SOFTNETWORKING 2018

[12] Adrian Hoban, Chair OSM TSC, Intel, Alfonso Tierno Sepulveda, OSM Resource Orchestration MDG, Telefónica, Gerardo García de Blas, OSM TSC, Telefónica, Kiran Kashalkar, OSM User Interface MDG, Rift.io, Mark Shuttleworth, OSM TSC, Canonical, Matt Harper, OSM TSC, Rift.io, Rajesh Velandy, OSM NW Service Orchestration MDG, Rift.io, OSM-Whitepaper-TechContent-ReleaseONE-FINAL, France, 2016 <https://osm.etsi.org/images/OSM-Whitepaper-TechContent-ReleaseONE-FINAL.pdf>

[13] Gianpietro Lavado , Achieving-end-to-end-NFV-with-OpenStack-and-Open-Source-MANO, Vancouver <https://www.openstack.org/assets/presentation-media/Achieving-end-to-end-NFV-with-OpenStack-and-Open-Source-MANO.pdf>

[14] Open Source MANO documentation master file, created by sphinx-quickstart on Wed Nov 20 23:11:37 2019, https://osm.etsi.org/wikipub/index.php/Openstack_configuration#Add_openstack_to_OSM

[15] OSM VNF ONBOARDING GUIDELINES, France, 2019, https://osm.etsi.org/images/OSM_VNF_Onboarding_Guidelines_June_2019.pdf

[16] Gianpietro Lavado, OSM Hackfest – Session 7.1

Introduction to Proxy Charms, 2019, <http://osm-download.etsi.org/ftp/osm-6.0-six/8th-hackfest/presentations/8th%20OSM%20Hackfest%20-%20Session%207.1%20-%20Introduction%20to%20Proxy%20Charms.pdf>

[17] A. Ciobanu, C. Conțu, E. Borcoci, "Charms and Virtual Network Functions primitives Experiments using Open Source MANO framework"

[18] Service Requirements for the 5G System Stage 1 (Release 16), 2018.

[19] N. Panwar, S. Sharma, and A. K. Singh, “A Survey on 5G: The Next Generation of Mobile Communication *,” 2015. Accessed: May 06, 2021. [Online]. Available: <http://www.surrey.ac.uk/5gic/research>