

TELEKOMUNIKAZIO TEKNOLOGIAREN INGENIARITZAKO  
GRADUA

# GRADU AMAIERAKO LANA

**5G SAREETAN VNF FUNTZIOAK MODU**

**OPTIMOAN KOKATZEKO SOFTWARE BATEN**

**DISEINUA ETA INPLEMENTAZIOA**



**Ikaslea:** Longo Bilbao, Aratz

**Zuzendaria:** Zabala Alberdi, Luis

**Ikasturtea:** 2023-2024

**Data:** Bilbon 2024ko uztailaren 26-an

# Laburpena

Telekomunikazioen esparruan, 5G sareen hedapena aurrerapen erabakigarria da gero eta handiagoak diren konektibitate- eta zerbitzu-kalitatearen eskaerei erantzuteko. Sare horien eraginkortasuna eta malgutasuna maximizatzeko funtsezko teknologietako bat sare-funtzioen birtualizazioa da (NFV), baliabide konputazionalak modu optimoan kudeatzeko aukera ematen baitu sare-funtzioak modu birtualizatuan inplementatzean. Ikuspegi hori bereziki garrantzitsua da misio kritikoko zerbitzuetarako, horietan fidagarritasuna eta azkartasuna funtsezkoak baitira komunikazioen segurtasuna eta errendimendua bermatzeko.

Proiektu honen helburu nagusia 5G sareetan VNF placement optimoa egiteko gai den software bat diseinatzea eta inplementatzea da, misio kritikoko zerbitzuen kalitatea hobetzeko. Solver-etan oinarritutako optimizazio-soluzioak integratuz eta erabiltzaile-interfazerako tresna ezberdinak arakatzuz, baliabideen erabilera minimizatzea eta sareen eraginkortasun operatiboa maximizatzea helburu duen sistema bat garatu da. Gainera, sare horietan NFVaren aplikazioa hobetu dezaketen metodologia eta teknologia desberdinen azterketa sakona egin da.

# Resumen

En el ámbito de las telecomunicaciones, el despliegue de las redes 5G representa un avance crucial para satisfacer las demandas crecientes de conectividad y calidad de servicio. Una de las tecnologías clave para maximizar la eficiencia y flexibilidad de estas redes es la virtualización de funciones de red (NFV), que permite gestionar de manera óptima los recursos computacionales al implementar funciones de red de manera virtualizada. Este enfoque es especialmente relevante para los servicios de misión crítica, donde la fiabilidad y la rapidez son esenciales para garantizar la seguridad y el rendimiento de las comunicaciones.

El presente proyecto tiene como objetivo principal diseñar e implementar un software capaz de realizar un VNF placement óptimo en redes 5G, para mejorar la calidad de los servicios de misión crítica. A través de la integración de soluciones de optimización basadas en solvers y la exploración de herramientas diversas para la interfaz de usuario, se ha desarrollado un sistema que busca minimizar el uso de recursos y maximizar la eficiencia operativa de las redes. Además, se ha realizado un análisis exhaustivo de diferentes metodologías y tecnologías que podrían mejorar la aplicación de NFV en estas redes.

# Abstract

In the telecommunications field, the deployment of 5G networks is a crucial advancement to meet the growing demands for connectivity and service quality. A key technology in maximizing the efficiency and flexibility of these networks is Network Function Virtualization (NFV), which allows optimal management of computational resources by implementing network functions virtually. This approach is especially relevant for mission-critical services, where reliability and speed are essential to ensure communication security and performance.

The primary objective of this project is to design and implement software capable of performing optimal VNF placement in 5G networks to improve the quality of mission-critical services. By integrating optimization solutions based on solvers and exploring different tools for the user interface, a system has been developed that seeks to minimize resource usage and maximize the operational efficiency of the networks. Additionally, an exhaustive analysis of different methodologies and technologies has been conducted to enhance the application of NFV in these networks.

# Aurkibidea

---

1.	Sarrera .....	7
2.	Testuingurua.....	9
2.1.	NFV Teknologia.....	9
2.1.1.	NFV arkitektura.....	10
2.1.2.	Sare-funtzio birtualizatuak eta sare birtualizatuaren funtzio-osagaiak...	14
2.2.	Misio Kritikoko zerbitzuak.....	15
2.2.1.	MCPTT zerbitzuaren oinarriak eta problematika.....	16
2.2.2.	MCPTT zerbitzuaren egokitzapena 5G sarera: VNF Placement.....	18
3.	Lanaren helburuak eta irismena .....	21
4.	Lanak dakartzan onurak .....	24
4.1.	Onura teknikoak .....	24
4.2.	Onura ekonomikoak .....	24
4.3.	Gizarte-onurak.....	25
5.	Alternatiben analisia.....	27
5.1.	Nola exekutatu VNF placement .....	27
5.1.1.	Solver bat erabiliz.....	27
5.1.2.	Reinforcement Learning.....	29
5.2.	Erabiltzailearen interfazea garatzeko tresnak: Python, Tkinter.....	30
5.2.1.	Zergatik Python eta Tkinter:.....	31
5.2.2.	Garapen-tresnen beste aukera batzuk .....	31
5.3.	Optimizazio-metodoen aukerak.....	31
5.3.1.	Trust-Constr.....	32
5.3.2.	Optimizazio-metodoen beste aukera batzuk.....	32
6.	Proposatutako irtenbidearen deskribapena.....	35

6.1.	Eredu matematikoa.....	35
6.1.1.	Sare topologia.....	35
6.1.2.	Trafiko klaseak .....	36
6.1.3.	VNF funtzioak .....	38
6.1.4.	Taldeak .....	39
6.1.5.	Ereduaren hasierako datuak.....	40
6.1.7.	Optimizazio-problema .....	42
6.2.	Eredu ebazpenerako software aplikazioa .....	44
6.2.2.	Erabilitako teknologiak.....	46
6.2.3.	Backend .....	48
6.2.4.	Frontend.....	54
6.2.5.	Datu-basearekin konexioa .....	71
6.2.6.	Direktorioen eta fitxategien egitura.....	73
6.2.7.	Lortutako emaitzak .....	75
7.	Lan-plana.....	84
7.1.	Lantaldea eta baliabide materialak .....	84
7.2.	Lan-paketeak eta zereginak .....	84
7.3.	Gantt diagrama.....	86
8.	Aurrekontua.....	88
8.1	Giza baliabideak .....	88
8.2.	Baliabide materialak .....	88
9.	Ondorioak.....	90
10.	Bibliografia .....	91

# 1. Sarrera

---

Telekomunikazio-sareek bilakaera esanguratsua izan dute hasieratik gaur egunera arte. Lehen belaunaldietatik (1G), ahots-dei analogikoak baino onartzen ez zituztenak, komunikazio digitalak eta testu-mezuak sartu zituen 2Gtik, abiadura handiko Internet mugikorrerako sarbidea eta aplikazio mugikorren ugaritzea bultzatu zuten 3G eta 4Graino. Gaur egun, 5G teknologia aurrerapen iraultzailea da, datu-abiadura oso altuak, latentzia izugarri baxuak eta gailu kopuru masibo bat modu eraginkorrean konektatzeko gaitasuna emanez. Jauzi teknologiko horrek, azken erabiltzailearen esperientzia hobetzeaz gain, aplikazio eta zerbitzu berriak ahalbidetzen ditu automobilgintza, osasuna, industria eta segurtasun publikoa bezalako sektoreetan.

5G teknologiaren helburu handienetako bat egungo komunikazio-ekosistema hobetzea da eta horretarako, 5G sareak optimizatzeko aurrerapen garrantzitsuenetako bat Software bidez Definitutako Sare-funtzioak (Software Defined Networking, SDN) eta Sare Funtzioak Birtualizatzeko (Network Functions Virtualization, NFV) estandarrik bezalako teknologiak sartzea da. SDNk sareko azpiegituraren kudeaketa malguagoa eta dinamikoagoa ahalbidetzen du, kontrol-planoa datu-planotik bereiziz, eta horrek programazio zentralizatua eta sareko trafikoaren optimizazioa errazten ditu. Bestalde, NFVk bereizi egiten ditu hardware sare espezifikoaren funtzioak, eta aukera ematen du funtzio horiek zerbitzari birtualizatu estandarretan exekutatzeko. Biek batera, SDNk eta NFVk plataforma arinagoa eta eraginkorragoa eskaintzen dute, baita sarearen funtzionaltasuna malgutasunez eta ekonomikoki hedatzea, funtsezkoak baitira 5Gren eskakizunak betetzeko.

Egoera hau bereziki interesgarria da latentzia izugarri baxua eta fidagarriak izateko beharra duten komunikazioetarako. Adibidez, Misio Kritikoko (Mision Critical, MC) zerbitzuak erabiltzen direnean. Zerbitzu hauek funtsezko aplikazio eta sistemak dira, eta fidagarritasun handia, latentzia txikia eta etengabeko erabilgarritasuna eskatzen dute, pertsonen segurtasuna eta ongizatea bermatzeko. Zerbitzu horiek funtsezkoak dira hainbat sektoretan, hala nola segurtasun publikoan, larrialdiko arreta medikoan, hondamendien kudeaketan eta beste egoera batzuetan, non komunikazio eta kontrol eraginkorrak bizitzaren eta heriotzaren arteko aldea ekar dezaketen. Zerbitzu horien

adibide bat Push to Talk Misio Kritikoa (MCPTT) da. Irrati bidezko komunikazio tradizionalen bilakaera da, eta erabiltzaileei aukera ematen die berehalako taldeko deiak egiteko, *walkie-talkie* erara, telekomunikazio-sareen gainean. 5G sareetan MCPTT ezartzeak zenbait erronka ditu, hala nola latentzia baxua, erabilgarritasun handia eta ingurune mugikor eta dinamikoetan zerbitzuaren kalitatea bermatzea. Dokumentu honetan, Sareko Funtzioen Birtualizazioak (NFV) erronka horiei nola hel diezaiekeen aztertuko da, VNF funtzioen kokapena eta funtzionamendua optimizatuz, 5G eta Beyond 5G sareetako MCPTT zerbitzuen eraginkortasuna eta fidagarritasuna hobetzeko.

Lan honetan, 5G sarearen bitartez eskaintzen den MC zerbitzu birtualizatua irudikatzen duen eredu matematiko bat hartzen da eta eredu horren soluzioa ematen duen software aplikazioa garatuko da.

Horretarako, aplikazio bat garatuko da. Alde batetik, aplikazioaren backend-a izango den programa garatuz. Lehen aipatutako VNF funtzioen kokapena definitzea izango du helburu. Eta bestetik, aplikazio honen erabilera egokirako eta programaren garapena erabiltzaile orok jakin behar ez izateko, frontend-a izango den erabiltzaile interfaze (User Interface, UI) simple bat garatuz.

Txostenaren hasieran, proiektuaren testuingurua ezarri da, eta 5G sareek eta sare-funtzioen birtualizazioak (NFV) telekomunikazio modernoek duten garrantzia nabarmendu da. Ondoren, helburu nagusiak eta bigarren mailakoak zehazten dira, baita proiektuaren irismena ere, 5G sareen testuinguru zabalenean duen garrantzia azpimarratuz. Horretaz gain, VNF placement-aren optimizazioak sare-operadoreentzat eta azken erabiltzaileentzat eskain ditzakeen onura tekniko, ekonomiko eta sozialak aztertzen dira. Alternatiben analisiaren atalean, hainbat metodologia eta teknologia ebaluatzen dira, eta proposatutako soluzioaren aukeraketa justifikatzen da. Jarraian, garatutako softwarea, haren arkitektura eta funtzionaltasuna xehetasunez deskribatuko dira, baita lan-planaren etapak eta baliabideak ere. Aurrekontuari dagokion atalean, proiektuari lotutako kostuen azterketa ere sartu da. Azkenik, txostenaren amaieran, lorpenak, aurrez aurre jarritako erronkak eta planteatutako konponbidea hobetzeko etorkizuneko bideak laburbiltzen dira, proiektuaren eraginari eta etorkizuneko aplikazioetarako duen potentzialari buruzko hausnarketa eskainiz.



## 2. Testuingurua

---

Lan honen esparrua behar bezala ezartzeko, atal honetan testuinguru garrantzitsua aurkezten da. Hasteko, NFV (Network Function Virtualization) teknologia eta 5G sareetan duen aplikazioa aztertu dira, eta sareko funtzioak birtualizatzeko funtsezko eginkizuna nabarmendu da, malgutasun eta efizientzia operatiboa hobetzeko.

Ondoren, misio kritikoko zerbitzuetaz hitz egingo da, eta 4G sareetatik 5G sareetara arteko bilakaeran zentratuko dira. Zerbitzu horiek azken belaunaldiko sare mugikorrek eskaintzen dituzten gaitasun berrietara egokitzea eztabaidatzen da, baita horiei lotutako erronkak eta problematikak ere, funtzio birtualizatuen posizionamendu optimoa barne (VNF placement), zerbitzuaren kalitatea eta aplikazio kritikoek eskatzen duten latentzia baxua bermatzeko.

Azkenik, proiektu honen testuinguruan funtsezko osagaiak inplementatzeko programazio-lengoaia gisa Python erabiltzea aztertzen da, softwarea garatzeko dituen ezaugarrien eta abantailen ikuspegi orokorra eskainiz.

### 2.1. NFV Teknologia

Sare-funtzioak birtualizatzeko teknologia (Network Virtualization Function, NFV) telekomunikazio-industria tradizionalaren mugei eta erronkei erantzuteko sortu zen. Testuinguru horretan, sareak sare-funtzio bakoitzerako ekipamendu espezifiko askorekin eraikitzen ziren, eta horrek azpiegitura oso espezializatua eta itxia sortzen zuen. Ikuspegi horrek hainbat zailtasun esanguratsu zekartzan sareko zerbitzuen hornitzaileentzat.

Lehenik eta behin, sareen konplexutasuna handitu egiten zen beharrezko ekipoen aniztasunagatik eta kantitateagatik, bakoitza funtzio espezifiko baterako fabrikatua eta diseinatua. Egoera horrek sarearen kudeaketa eta mantentzea zailtzeaz gain, kostu operatiboak eta kapital-kostuak ere areagotzen zituen. Zerbitzu berri bakoitzak ekipamendu berriak integratzea eskatzen zuen, eta, horretarako, azpiegitura fisiko gehigarria, elikadura elektrikoa, rack-etako espazioa eta instalatzeko eta jarduteko gaitasuna zuten langileak behar ziren.

Gainera, azpiegitura horren zurruntasunaren ondorioz, zerbitzu berriak sartzea prozesu motela eta nekagarria zen. Ekipo berriak egungo sarean integratzeko beharrak atzerapen nabarmenak eragiten zituen, eta hornitzaileek merkatuaren eskaerei azkar erantzuteko gaitasuna mugatzen zuten. Hori bereziki problematikoa zen bizi-ziklo laburreko zerbitzuetarako, non bilakaera teknologiko azkarrak esan nahi baitzuen ekipoak zaharkitu egin zitezkeela hasierako inbertsioa berreskuratu baino lehen.

Erronka horien aurrean, industria irtenbide malguagoak eta eraginkorragoak bilatzen hasi zen. Hemen sartzen da jokoan NFV. NFVak sareko funtzioak hardware espezifikoetik bereiztea ahalbidetzen du, zerbitzari estandarretan birtualizatuz eta exekutatzuz. Birtualizazio horrek hainbat abantaila eskaintzen ditu: hardware jabearen mendekotasuna murrizten du, sarearen kudeaketa eta eskalatzea errazten du, eta zerbitzu berrien implementazioa bizkortzen du, ekipo berriak fisikoki integratzeko beharra ezabatzen baitu. Gainera, baliabideen erabilera eraginkorragoa ahalbidetzen du, funtzio birtualizatuak birkokatu eta dinamikoki esleitu baitaitezke, sarearen eta zerbitzuaren beharren arabera.

### 2.1.1. NFV arkitektura

Sare-funtzioen birtualizazioaren arkitektura funtsezkoa da teknologia horrek telekomunikazio-sareak nola eraldatzen dituen ulertzeko. Funtsezko bi arkitektura mota daude NFVn: goi-mailako arkitektura eta ETSIk (European Telecommunications Standards Institute) argitaratutako erreferentziazko arkitektura.

#### ***Goi-mailako arkitektura***

NFVren goi-mailako arkitektura hiru elementu estandarizatutan banatzen da:

1. Sare-funtzio birtualizatuak (VNF)
  - a. VNFak sareko funtzioen software-inplementazioak dira, tradizionalki hardware dedikatuan exekutatzen zirenak. VNFen adibideak routerrak, firewall-ak eta karga-kulunkagailuak dira. Funtzio birtualizatu horiek NFVI azpiegitura birtualizatuaren gainean gauzatzen dira, sarean malgutasun eta eskalagarritasun handiagoa ahalbidetuz.

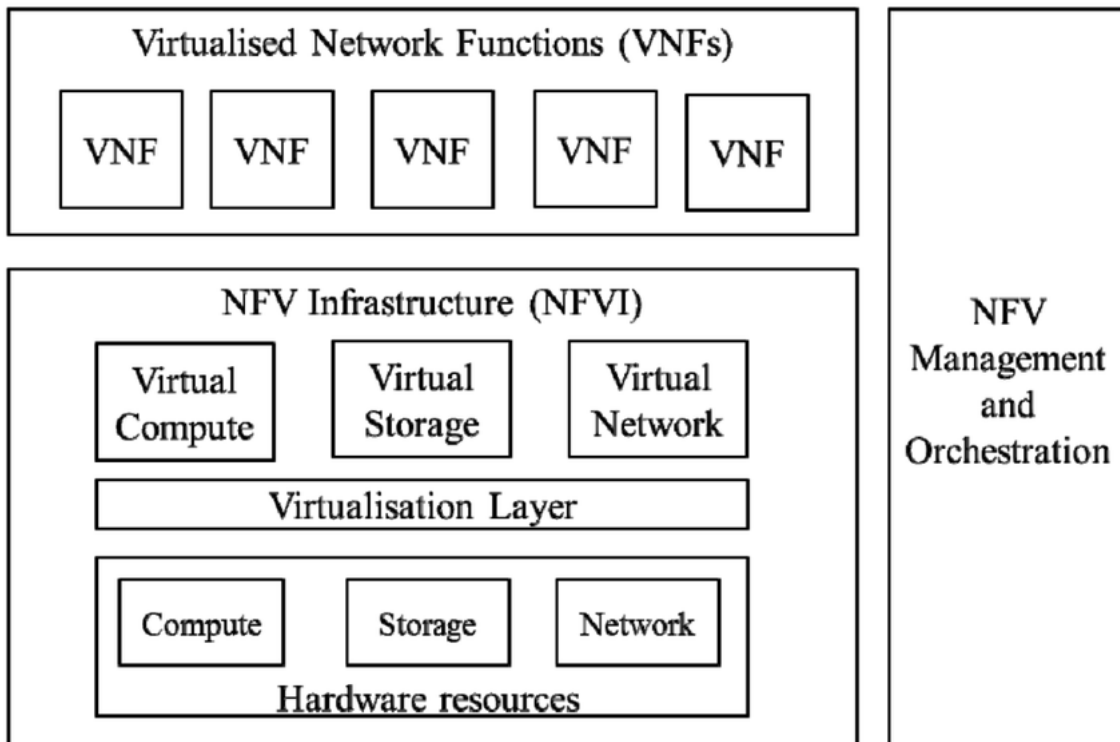
2. Sare-funtzioak birtualizatzeko azpiegitura (NFVI)

- a. NFVI VNFak exekutatzen diren ingurunea da. Ingurune horretan, birtualizatu diren baliabide fisiko guztiak sartzen dira (hala nola zerbitzariak, biltegiak eta sareak). NFVIak abstrakzio-geruza bat ematen du, VNFei azpiko hardwaretik aparte lan egiteko aukera ematen diona.

3. NFVren kudeaketa eta orkestrazioa (NFV MANO)

- a. NFV MANO VNF eta NFVI baliabideen bizi-zikloaren kudeaketaren eta kontrolaren arduraduna da. VNFak eta azpiegitura birtualizatuak orkestratzeko, kudeatzeko eta gainbegiratzeko funtzioak barne hartzen ditu. NFV MANOk ziurtatzen du VNFak NFVI ingurunean modu eraginkorrean inplementatu, konfiguratu eta jarduten direla.

*Irudia 1: goi-mailako arkitektura. Iturria: [1]*



## ***ETSIk argitaratutako erreferentzia arkitektura***

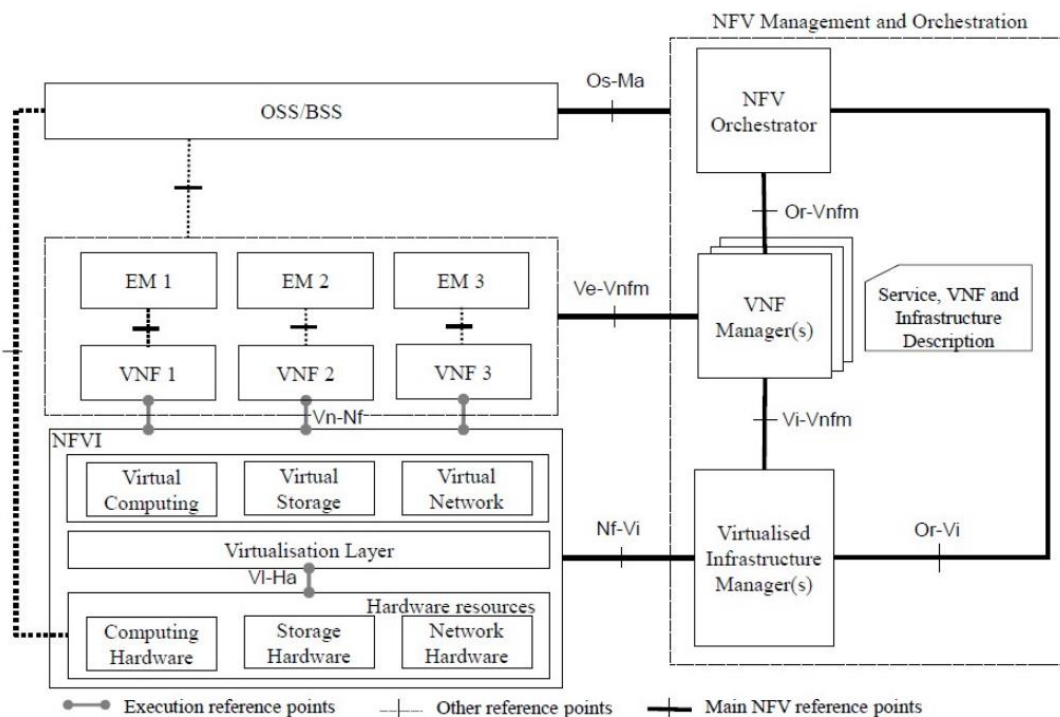
ETSIk garatutako NFVren erreferentziazko arkitektura lau elementu nagusitan egituratzen da:

1. Kudeaketa eta Orkestrazioa (MANO)
  - a. MANO hiru osagai erabakigarritan banatzen da:
    - i. NFV Orchestrator (NFVO): NFVO OSS/BSS (Operations Support Systems/Business Support Systems) sistemekin komunikatzen da eta beste kudeaketa-sistema batzuekin integratzea ahalbidetzen du. VNFMa eta VIMa kudeatzea eta kontrolatzea da haren funtzio nagusia, eta VNFen hedapenean eta eragiketan koherentzia ziurtatzea.
    - ii. VNF Managerra (VNFM): VNFMak NFVOk kontrolatutako VNFen bizi-zikloa kudeatzen du. VIMari VNFak inplementatu, konfiguratu eta desegiteko beharrezko jarraibideak bidaltzen dizkio.
    - iii. Virtualized Infrastructure Manager (VIM): VIMak NFVI azpiegitura kudeatzen du, baliabide birtualen erregistro zehatza mantenduz eta baliabide fisikoei esleituz. NFVIren barruko software- eta hardware-baliabideak kudeatzen ditu, eta erabilera optimoa eta baliabideen eskuragarritasuna ziurtatzen ditu.
2. Sare-funtzio birtualizatuak (VNF)
  - a. VNFak sareko funtzioen bertsio birtualizatuak dira, hala nola routerrak, firewall-ak eta sareko beste gailu batzuk. VNF bakoitza hardware- eta software-osagaietan banatzen da, eta bakoitzak bizi-ziklo bat du. VNFek makina birtual ugarran (VM) jardun dezakete, eta MANOren bidez kudeatzen dira. VNFen barruan, Element Management System (EMS) arduratzen da sareko elementuen konfigurazioa eta funtzionamendua

kudeatzeaz, akatsak, konfigurazioa, kontabilitatea, errendimendua eta segurtasuna kudeatzeko funtzioak barne (FCAPS).

3. Sare-funtzioak birtualizatzeko azpiegitura (NFVI):
  - a. NFVIak VNFak exekutzeko beharrezkoak diren baliabide birtualizatu guztiak biltzen ditu. Horren barruan daude prozesamendu birtuala, biltegiatze birtuala eta sare birtualak. Hiperbisorea, VirtualBox bezala, funtsezko pieza bat da, baliabide fisiko horien birtualizazioa ahalbidetzen duena, VNF bakoitzerako bereizitako ingurune birtualak eskainiz.
4. Eragiketa- eta negozio-euskarrien sistemak (OSS/BSS):
  - a. OSS/BSS sistema kritikoak dira, eta kontabilitateaz, fakturazioaz eta eragiketa- zein merkataritza-euskarrien beste funtzio batzuek arduratzen dira. Sistema horiek NFVorekin elkarrengaitan dute, zerbitzuak merkataritza- eta eragiketa-eskakizunen arabera inplementatzen eta kudeatzen direla ziurtatzeko.

**Irudia 2: ETSIk argitaratutako erreferentzia arkitektura. Iturria: [2]**



## 2.1.2. Sare-funtzio birtualizatuak eta sare birtualizatuaren funtzio-osagaiak

Sare-funtzio birtualizatuak (VNF) eta sare birtualizatuaren funtzio-osagaiak (VNFC) instantziatzea eta sortzea funtsezko prozesuak dira NFVren esparruan. Prozesu horiek ziurtatzen dute sareko funtzioak konfiguratuta eta prest egotea ingurune birtualizatuan eraginkortasunez jarduteko.

### ***VNF baten instantziazioa***

VNF baten instantziazioa VNF deskribatzailean (VNFD) oinarritzen da, zeinak sarearen funtzioaren ezaugarriak eta eskakizunak definitzen baititu. VNFDak zehazten du nola sortzen, konfiguratzen eta amaitzen diren VNF instantziak.

VNF bakoitza hainbat VNFCk osatzen dute, oro har lauk. Horietako bat nagusia da eta besteak bigarren mailakoak dira. VNF bat sortzeko prozesua hasteko, hasierako baldintza hauek bete behar dira:

- VNFak ezarrita egon behar du.
- VNF Managerrak (VNFM) martxan egon behar du.
- Element Management System (EMS) martxan egon behar da.

Behin baldintza horiek beteta, VNF instantziazioa egin daiteke. Prozesua VNF "Instantiated Not Configured" egoeran dagoenean amaitzen da; horrek esan nahi du VNF sortu dela, baina oraindik ez dagoela konfiguratuta jarduteko.

### ***VNFC baten instantziazioa***

VNFC instantzia bat sortzeak VNF instantziaren antzeko prozesu bat jarraitzen du, zenbait espezifikotasunekin:

- Hasierako baldintzak bat datoz VNF bat sortzearekin: ezarritako VNF, VNFM operatiboa, eta zerbitzuan dagoen EMS.

- VNFMa VNFC instantziaren konfigurazioa kudeatzeaz arduratzen da. Horren barruan sartzen da makina birtualerako (VM) beharrezko konfigurazioa sartzea eta prozesuak abiaraztea.
- VMa konfiguratu ondoren, VNFari edo EMSari jakinarazten zaio VMa konfiguratzeko prest dagoela.
- Prozesua osatzen da VNFC instantzia sortuta, konfiguratuta eta funtzionatzeko prest dagoenean.

### ***VNFC instantzien amaitzea***

VNFC instantzien amaierak ere ondo definitutako urrats multzo bat jarraitzen du:

- Hasierako baldintzek VNFCa sortuta eta funtzionamenduan egotea eskatzen dute.
- VNFmak amaitzeko eskaera egiten du. Prozesu horretan, VNFCaren egoera gordetzeko aukera dago, beharrezkoa bada.
- Prozesua osatzen da VNFCk erabilitako baliabideak hautatzen eta askatzen direnean.

### ***VNF instantzien amaitzea***

VNF instantziak hasierako baldintza hauen arabera amaitu daitezke:

- VNFak "Not Configured" edo "Instantiated Configured" egoeran egon behar du.

VNFmak bi aukera ditu instantzia amaitzeko: VNFCk zehaztutako jarraibideei jarraitzea edo amaitzeko eskaera zuzen bat VNFra bidaltzea. Prozesua amaitzean, VNF instantzia ez da existitzen, eta lotutako baliabideak askatu egiten dira [2].

## **2.2. Misio Kritikoko zerbitzuak**

Misio kritikoko zerbitzuak egoera kritikoetan eta larrialdietan komunikazio fidagarriak, seguruak eta lehentasunezkoak eskaintzeko diseinatutako telekomunikazio-aplikazioen kategoria espezializatu bat dira. Zerbitzu horiek funtsezkoak dira segurtasun publikoaren, larrialdi-zerbitzuen eta hondamendien aurrean erantzuteko; izan ere, bat-bateko komunikazio fidagarria funtsezkoa da ekintzak koordinatzeko eta bizitzak salbatzeko.

Telekomunikazio-sareen bilakaerak, bereziki 4G LTE eta orain 5G bezalako abiadura handiko teknologia mugikorren sarrerarekin eta hedapenarekin, gaitasun eta aukera berriak gaitu ditu misio kritikoko zerbitzuak hobetzeko. Sare horiek, datuen abiadura eta gaitasun handiagoa izateaz gain, funtzio aurreratuak integratzea ahalbidetzen dute, hala nola IPren gaineko ahotsa (VoIP), bideoa denbora errealean eta datu kritikoak ingurune integratu batean.

Hala ere, sare mugikorretan misio kritikoko zerbitzuak eraginkortasunez ezartzeak erronka handiak ditu. Latentzia txikiko bermea, trafiko kritikolari lehentasuna ematea, segurtasun sendoa eta erabilgarritasun handia funtsezkoak dira zerbitzu horien eskakizun operatiboak betetzeko. Gainera, sare-funtzioak birtualizatzeak (NFV) eta ertzean konputatzeak (MEC) aukera berriak eskaintzen dituzte ingurune mugikorretan misio kritikoko zerbitzuen entrega optimizatzeke eta hobetzeko.

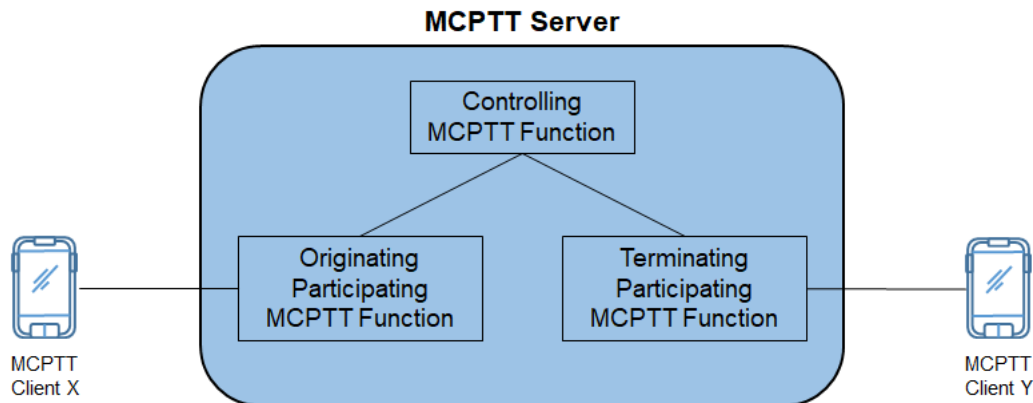
Horrela bada, atal honetan aztertuko da nola 5G sareak eta *Beyond 5G* misio kritikoko zerbitzuak diseinatzeko eta hedatzeko modua eraldatzen ari diren, bereziki segurtasun, larrialdi eta zerbitzu kritikoetako erakundeek erabiltzen dituzten push to talk (PTT) aplikazioetan.

### 2.2.1. MCPTT zerbitzuaren oinarriak eta problematika

Misio Kritikoaren Push To Talk (MCPTT) zerbitzua funtsezkoa da segurtasun-endarrentzat, suhiltzaileentzat, larrialdi medikoetarako zerbitzuentzat eta erreskate-taldeentzat, eta 3GPPk bultzatutako misio kritikoko zerbitzu garrantzitsuenetako bat da. Gaur egun, MCPTT zerbitzari zentralizatu baten bidez bermatzen da. Zerbitzari horrek 3GPPk definitutako estandarrari jarraitzen dio, eta MCPTT AS Participating eta MCPTT AS Controlling rolak betetzen ditu. Eredu horretan, zerbitzuaren funtzionalitate guztiak zerbitzari zentral batetik ematen dira, eta komunikazio guztiak zerbitzari zentraletik igarotzen dira. Bezeroek MCPTT aplikazioaren transakzio bakoitzean zerbitzariaren partaidetza-funtzioarekin elkarreragiten dute.



### Irudia 3: MCPTT zerbitzaria.



Ikuspegi zentralizatu horrek, funtzionala izan arren, hainbat erronka ditu. Kontrol-planoaren eta erabiltzailearen planoaren komunikazio guztiek zerbitzari zentraletik igaro behar dutenez, latentzia-arazoak sor daitezke, batez ere sareko kongestio kondizioetan edo kondizio suoptimoetan. Latentzia-arazo horiek errendimenduaren funtsezko adierazleak (KPI) ez betetzea eragin dezakete, komunikazio kritikoen fidagarritasunari eta eraginkortasunari eraginez.

MCPTTki bi funtzionamendu modu eskaintzen ditu: sarearekin eta sarerik gabe. 5G sareen bidezko funtzionamenduaren kasuan, MCPTT zerbitzua IP bidez irisgarria den zerbitzari zentralizatu gisa inplementatzen da, 5G azpiegituran aplikatzekoak diren beste zerbitzu batzuen antzera. Zerbitzuaren koordinazioak IMS seinaleztapen-azpiegitura (IP Multimedia Subsystem) eskatzen du, 5G operadoreek aplikazio-zerbitzuetarako deiak (AS) eta komunikazioetan parte hartzen duten erabiltzaile-gailuen konexioa kudeatzeko erabiltzen dutena. Nabarmentzekoa da MCPTTki banakako zein taldeko komunikazioak ahalbidetzen dituela [3].

5G sareen etorrerarekin eta haratago (Beyond 5G), MCPTT zerbitzua hobetzeko aukera berriak irekitzen dira Sarearen Ertzeko Konputazioan (Mobile Edge Computing, MEC) oinarritutako soluzioen bidez. Sare horietan, tradizionalki zentralizatuta egon diren funtzionalitate batzuk sareko funtzio birtualizatu gisa inplementatu eta sare mugikorraren muturretara eramán daitezke. Ikuspegi deszentralizatu horrek ahalbidetzen du zerbitzuak azken erabiltzaileengandik gertuago egotea, latentzia murriztuz eta zerbitzuaren kalitatea hobetuz.

Gainera, MCX Misio Kritikoko zerbitzu birtualizatu batek (PTT, MCData eta MCVideo barne) onura handiak izan ditzake VNF funtzioen hedapen adimentsuaz. Hedapen adimentsu horrek sareko baliabideen erabilera optimizatzeaz gain, funtzionaltasun kritikoak erabiltzaileengandik gertuago, eskuragarri eta operatibo egotea ere ziurtatzen du, erresilientzia eta erantzun handiagoa bermatuz larrialdi-egoeretan. Hobekuntza horiek 5G eta B5G sareen testuinguruan ezartzea, beraz, logikoa ez ezik, beharrezkoa ere bada misio kritikoko zerbitzuen gero eta eskakizun handiagoak betetzeko [4].

## 2.2.2. MCPTT zerbitzuaren egokitzapena 5G sarera: VNF Placement

VNF placement sareko azpiegitura baten barruan VNFak hedatzeko kokaleku egokienak hautatzeko prozesuari dagokio. VNFen placement (kokapena) alderdi kritikoa da sareen kudeaketan, bereziki 5G sareen gaineko Push To Talk Misio Kritikoko (MCPTT) zerbitzuen testuinguruan. Sare horietan VNFen placement optimizatzea funtsezkoa da errendimendu egokia bermatzeko, sarearen delay (atzerapena) minimizatuz eta larrialdi-egoeretan erantzun azkarra eta fidagarria ziurtatuz.

MCPTTren testuinguruan, zerbitzuek latentzia baxuko eta erabilgarritasun altuko baldintza zorrotzak bete behar dituzte. Izan ere, komunikazioaren edozein atzerapenek ondorio kritikoak izan ditzake segundo bakoitzak kontatzen duen larrialdi-egoeretan. Beraz, VNFen placement-a kontu handiz planifikatu eta optimizatu behar da, delay-a minimizatzeke eta sarearen errendimendua maximizatzeke.

MCPTT sareetan VNFen placement optimizatzeke, hainbat estrategia erabil daitezke:

- Erabiltzaileekiko hurbiltasuna:
  - VNFak azken erabiltzaileengandik ahalik eta hurbilen jartzeak latentzia nabarmen murriztu dezake. Hori lortzeko, Konputazio-arkitekturak erabil daitezke Sarearen Ertzean (MEC), non VNFak sare mugikorraren muturretan kokatutako sareko nodoetan hedatzen diren, erabiltzaileen gailuetatik gertuago.
- Banaketa geografikoa:

- VNFak geografikoki banatzeak erresilientzia eta eskuragarritasun handiagoa ahalbidetzen du. MCPTT sareetan, funtsezkoa da funtzio kritikoak beti eskuragarri egotea, baita sare-akatsen edo gainkargen kasuan ere. Banaketa geografiko egoki batek ziurtatzen du zerbitzuak azkar birbidera daitezkeela beste VNF batzuetara, behar izanez gero.
- Karga balantzea:
  - Karga balantzeatzeko mekanismoak ezartzeak trafikoa hainbat VNFren artean modu ekitatiboan banatzen lagun dezake, pilaketa-puntuak saihestuz eta sarearen errendimendu orokorra hobetuz. Hori bereziki garrantzitsua da MCPTT sareetan, eskari handiko trafikoa puntuak ezin baitira aurreikusi eta handiak izan baitaitezke.
- Latentzia oinarritutako optimizazioa:
  - Latentzia VNFen placementerako funtsezko faktoretzat hartzen duten optimizazio-algoritmoak erabiltzea. Algoritmo horiek sareko ibilbideak azter ditzakete eta VNFetarako kokapen optimoak aukera ditzakete, atzerapena minimizatuz eta komunikazioaren eraginkortasuna hobetuz.

5G sareetan VNFen placement estrategiak ezartzeak hainbat abantaila esanguratsu eskaintzen ditu Push To Talk Misio Kritikoaren (MCPTT) zerbitzuen errendimendua optimizatzeko. 5G sareek datu-tasa altuagoak eta gailu-dentsitate handiagoa jasateko duten gaitasunak MCPTT zerbitzuen inplementazio eraginkorra errazten du, latentzia murriztuz eta zerbitzuaren kalitatea hobetuz.

Lehenik eta behin, 5G sareen malgutasun handiagoak, softwarean oinarritutako arkitekturarekin eta birtualizazio-gaitasunarekin, malgutasun handiagoa ematen dio VNFen placement dinamikoari. Malgutasun horri esker, azkar egokitu daiteke sarearen baldintza aldakorretara eta erabiltzaileen eskarietara, eta zerbitzu kritikoak operatibo eta eraginkor mantendu ahal izango direla bermatzen da, baita eskari handiko egoeretan edo sareko akatsetan ere.

Gainera, Multi-Access Edge Computing (MEC) 5G arkitekturan integratzeak VNFak azken erabiltzaileengandik hurbilago egotea ahalbidetzen du. Sare mugikorraren

muturretan kokatutako sareko nodoetan VNFak hedatzean, latentzia nabarmen murriztu daiteke, sareko funtzioak erabiltzaileen gailuetatik gertuago exekutatzen baitira. Hurbiltasun hori bereziki garrantzitsua da MCPTT zerbitzuen testuinguruan, non komunikazioan edozein atzerapenek ondorio kritikoak izan ditzakeen.

5G sareetan karga balantzeatzeko mekanismoak ezartzeak MCPTT zerbitzuen errendimendua optimizatzen ere laguntzen du. Mekanismo horiek trafikoa hainbat VNFren artean modu ekitatiboan banatzen laguntzen dute, pilaketa-puntuak saihestuz eta sarearen errendimendu orokorra hobetuz. Azken finean, MCPTT sareetan trafiko-gailurrak aurreikusi ezinak eta handiak izan daitezke, eta funtsezkoa da larrialdi-egoeretan erantzun azkarra eta fidagarria bermatzea.

Azkenik, funtsezkoa da VNFen placementean latentzia funtsezko faktoretzat hartzen duten optimizazio-algoritmoak erabiltzea. Algoritmo horiek sareko ibilbideak azter ditzakete eta VNFetarako kokapen optimoak aukera ditzakete.

### 3. Lanaren helburuak eta irismena

---

Atal honetan proiektuaren helburu nagusiak eta bigarren mailakoak zehaztuko dira, baita espero den irismena ere. Proiektuaren ibilbide argi bat ezartzea da funtsezko helburua, proposatutako softwarearen garapena eta inplementazioa gidatuko duten helburu nagusia eta bigarren mailako helburuak definituz.

Proiektu honen **helburu nagusia** software bat diseinatzea eta inplementatzea da, 5G sareetan VNF funtzioak modu optimoan kokatzeko. Era honetara, sarearen eraginkortasuna indartuz eta Misio Kritikoko zerbitzuen kalitatea hobetuz, funtzio horiek estrategikoki hedatzen direla ziurtatuz, sarearen errendimendua eta latentzia optimizatzeko.

Helburu nagusia lortzeko, bigarren mailako hainbat helburu zehaztu dira, eta horietako bakoitzak proiektuaren alderdi espezifikoak eta beharrezkoak jorratzen ditu. Bigarren mailako helburuak honako hauek dira:

- 1. Proposatutako eredu matematikoa ulertzea:** 5G sareetan VNFak jartzea optimizatzeko, problema deskribatzen duen eredu matematikoa sakon ulertu behar da. Helburu horrek berekin dakar tartean dauden formulak eta murrizketak aztertzea eta ezagutzea, hala nola sistemaren egonkortasuna, gaitasun konputazionala eta sare-latentziak. Funtsezkoa da ekuazio eta erlazio matematikoak behar bezala interpretatzea, softwareak bere funtzioa eraginkortasunez eta efizientziaz beteko duela ziurtatzeko. Gainera, ereduaren parametroek VNFen jardunean nola eragiten duten ulertzeak azken soluzioa hobetzeko doikuntza finak egitea ahalbidetuko du.
- 2. Lengoaia eta tresna berriei buruz ikastea:** bigarren helburua softwarea garatzeko lengoia ezberdinen trebetasun praktikoak eskuratzea da. Horrek esan nahi du: lengoia egokia aukeratzea, erabiltzaileei softwarearekin modu eraginkorrean elkarreragiteko aukera emango dietenak; lengoia hoiien oinarrizko kontzeptuak ulertzea; eta lengoia hoiien liburutegi gehigarriak esploratzea, proiektua garatzeko baliagarriak izan daitezkeenak.

### 3. Misio kritikoko zerbitzuen sareen funtzionamendua eta horien problematika

**aztertzea:** helburu hau zerbitzu horien euskarri diren sareek nola funtzionatzen duten aztertzean datza, haien ezaugarriak eta aurrez aurre dituzten erronkak identifikatuz. Ezagutza hori funtsezkoa da sare horien kalitatea eta fidagarritasuna hobetzeko softwarea diseinatzeko. Horretarako, honako alderdi hauek hartzen dira kontuan: aztertzea nola dauden egituratuta 5G sareak aplikazio kritikoak jasateko, eta zer alderdi teknologiko diren funtsezkoak haien funtzionamendurako, eta identifikatzea Misio Kritikoko sareek dituzten arazo komunak, hala nola sarearen pilaketa, baliabideen kudeaketa eta akatsen aurreko erresilientzia.

### 4. Softwarea implementatzea eta probatzea agertoki simulatuetan:

proiektuaren funtsezko zati bat garatutako softwarea baliozkotzea da. Helburu horrek honako hau eskatzen du: sareko baldintzak eta trafiko-mailak islatzen dituzten ingurune simulatuak garatzea, eta softwarearen eraginkortasuna neurtzea latentziari, baliabideen erabilerari eta egonkortasunari dagokienez, hainbat agertokitan.

Proiektuaren irismena hura garatzeko muga eta murrizketek definitzen dute. Beraz, proiektuaren irismenerako puntu hauek definitzen dira:

- **Prototipo egoki bat garatzea:** 5G sareetan VNFren kokapen optimoa frogatzen duen prototipo funtzional bat garatzea, horretarako behar diren tresna egokiak hautatuz eta diseinu-erabaki aproposak hartuz.
- **Misio kritikoko zerbitzuen optimizazioan laguntzea:** garatutako softwareak Misio Kritikoko zerbitzuetan komunikazioen kalitatea eta eraginkortasuna hobetzeko duen eragina ebaluatzea.
- **Eredu matematikoa baliozkotzea:** probak eta analisiak egitea, proposatutako eredu matematikoa behar bezala inplementatzen dela eta ezarritako helburuak betetzen dituela baieztatzeko.

Aipatzekoa da, etorkizunera begira, oso onuragarria izango litzatekeela optimizazio-prozesuetan adimen artifizialaren integrazioa aztertzea. Adimen artifizialak ikasteko eta egokitze gaitasun aurreratuak ekar ditzake, eta horrek nabarmen hobetu dezake VNF funtzioak ezartzeko eta baliabideak denbora errealean kudeatzeko eraginkortasuna. Ikuspegi horrek konponbide egokitzaileagoak eta eraginkorragoak ahalbidetu ditzake sare

konplexu eta aldakorretan, eta hori aurrerapauso garrantzitsua izango litzateke Misio Kritikoko eragiketarako. Testuinguru horretan IAren teknikak ezartzeak, 5G sareen eraginkortasuna bultzatzeaz gain, aplikazio adimentsuagoak eta autonomoagoak garatzeko aukera berriak irekiko lituzke.

## 4. Lanak dakartzan onurak

---

Atal honetan, proiektu honen garapenak hainbat ikuspegitatik ekar ditzakeen onurak aztertzen dira. Helburua da Misio Kritikoko zerbitzuetarako 5G sareen optimizazioak hobekuntza teknikoak, ekonomikoak eta sozialak nola eragin ditzakeen identifikatzea eta deskribatzea. Azterketa hori hiru azpiataletan egituratuta dago: onura teknikoak, onura ekonomikoak eta onura sozialak.

### 4.1. Onura teknikoak

Misio Kritikoko zerbitzuetarako 5G sareak optimizatzeko onura tekniko esanguratsuak eskaintzen ditu. Onura horiek komunikazio kritikoko sareen kudeaketan eta eragiketan parte hartzen duten enpresetara, gobernu-erakundeetara eta beste eragile batzuetara zabaltzen dira.

Onura tekniko nagusien artean sarearen errendimendua hobetzea dago; izan ere, latentzia murriztea funtsezkoa da denbora errealean erantzunak behar dituzten Misio Kritikoko aplikazioetarako, eta gaitasuna eta eraginkortasuna handitzeak aldi bereko konexio gehiago ahalbidetzen ditu. Fidagarritasuna eta erresilientzia ere hobetzen dira, zerbitzuaren jarraitutasuna ziurtatuz, baita eskaera handiko egoeretan edo sareko akatsetan ere, eta akatsen kudeaketa hobea erraztuz, sarea azkar eta eraginkortasunez berrkonfiguraturaz.

Beste alderdi garrantzitsu bat eskalagarritasuna da, aukera ematen baitu eskaera berrietara egokitzeko eta zerbitzu berriak sartzeko, errendimendua eta kalitatea mantenduz. Azkenik, segurtasuna areagotu egiten da funtzio kritikoak ingurune seguruetan gauzatzen direla ziurtatzean, komunikazioak mehatxuen eta erasoen aurka babestuz.

### 4.2. Onura ekonomikoak

5G sareetako VNF funtzioak optimizatzeko, errendimendu teknikoa hobetzeaz gain, onura ekonomiko garrantzitsuak ere eskain diezazkieke tartean dauden erakundeei, hala nola telekomunikazioetako operadoreei eta beste zerbitzu-hornitzaile batzuei.



Onura ekonomiko nagusietako bat kostu operatiboak murriztea da. Horretarako, eraginkortasun energetiko handiagoa lortzen da, beharrezkoa ez den baliabideen kontsumoa minimizatzen baitu, eta mantentze-lan eta kudeaketa eraginkorragoak egiten dira, horri lotutako kostuak murrizteko. Gainera, inbertsioaren itzulkina maximizatzea lortzen da (ROI); izan ere, dagoen azpiegitura hobeto erabiltzeak aukera ematen die operadoreei 5G sareetan egiten dituzten inbertsioak maximizatzeko eta zerbitzu-ahalmena handitzeko, inbertsio gehigarri handirik gabe.

Funtsezko beste onura ekonomiko bat merkatuko lehiakortasuna da. Bertan, latentzia txikiko eta fidagarritasun handiko kalitate handiko zerbitzuak emateak desberdintzen ditu operadoreak eta lehiakideak, bezero gehiago erakarriz eta diru-sarrerak handituz. Gainera, funtzio eta zerbitzu berriak eraginkortasunez ezartzeko gaitasunari esker, operadoreek berrikuntzaren abangoardian jarraitzen dute telekomunikazioen sektorean.

### 4.3. Gizarte-onurak

5G sareetan VNFen ezarpena hobetzeak ere eragin nabarmena du gizartean, bereziki Misio Kritikoko zerbitzuen testuinguruan, sarritan hil edo biziko egoerez arduratzen baitira.

Gizarte-onura garrantzitsuenetako bat larrialdiei ematen zaien erantzuna hobetzea da; izan ere, latentzia murrizteak eta sarearen fidagarritasuna hobetzeak larrialdi-zerbitzuei aukera ematen diete egoera kritikoei azkarrago erantzuteko, biziak salba baititzakete. Gainera, koordinazio eta komunikazio hobea lortzen da hainbat agentzia eta larrialdi-zerbitzuren artean, erantzunaren eraginkortasuna handituz. Segurtasun publikoa ere onuragarria da; izan ere, segurtasun-indarren eta beste zerbitzu publiko batzuen komunikazioak azkarrak eta seguruak izango direla bermatzea funtsezkoa da herritarrak babesteko, eta hondamendi naturalen edo eskala handiko gorabeheren egoeretan, sare optimizatu batek ziurtatzen du komunikazio kritikoei funtzionatzen jarraituko dutela, erreskate- eta laguntza-eragiketarako erraztuz.

Azkenik, 5G sareen optimizazioak bizi-kalitatea hobetzen laguntzen du, osasun- eta segurtasun-zerbitzuetarako sarbide hobea ahalbidetuz, eta horrek komunitateetako bizi-

kalitatea hobetzen du eta inklusio digitala sustatzen du, pertsona gehiagok zerbitzu kritikoetarako eta aukera ekonomikoetarako sarbidea izatea ziurtatuz.

## 5. Alternatiben analisisa

---

Lan honen muina den software aplikazioa azaldu aurretik, beharrezkoa da proiektua egiteko kontuan hartu behar diren diseinu-alternatibak aurkeztea. Proiektuaren diseinua egiterakoan, erabakiak hartu behar dira, arrakasta edo porrota ekar dezaketenak. Horregatik, dauden alternatibak zehatz-mehatz aztertzea komeni da. Ez da beharrezkoa aukera guzti-guztiak kontuan hartzea, baina bai irtenbide bat emateko orduan kritikoak izan daitezkeenak.

Atal honetan, alternatiba-azterketa hori egingo da, kasu bakoitzean egokiena zein den zehazteko. Hain zuzen ere, lehenengo, *VNF placement* aukerak planteatzen dira; ondoren, erabiltzaile-interfazea garatzeko tresna desberdinak; eta, azkenik, optimizazio-metodoen artean hautaketa bat egiten da.

*VNF placement* gauzatzeko bi ikuspegi nagusi ebaluatuko dira: *solver* bat erabiltzea eta *Reinforcement Learning* erabiltzea. Halaber, erabiltzailearen interfazea garatzeko erabiltzen diren tresnak eztabaidatuko dira, hala nola Python eta Tkinter. Amaitzeko, *minimize* funtzioaren barruan, programaren exekuzio eraginkorra ziurtatzeko erabil daitezkeen optimizazio-metodoak aztertuko dira.

### 5.1. Nola exekutatu VNF placement

Nagusiki bi aukera daude VNF placement egiteko: *solver* bat erabiliz edo Reinforcement Learning bidez. Ikuspegi bakoitzak bere abantailak eta desabantailak ditu, eta horiek zehatz-mehatz aztertuko dira atal honetan.

#### 5.1.1. Solver bat erabiliz

Solver bat optimizazio-problema konpontzen dituen tresna matematiko edo software bat da, eta funtzio objektibo baten balio optimoa aurkitzen du murrizketa-multzo baten barruan. VNF placement delakoaren testuinguruan, sare jakin batean sare-funtzio birtualizatuen (VNF) esleipen optimoa zehazteko erabiltzen dira solverrak.

Solverrak bi kategoriatan banatzen dira nagusiki: *solver komertzialak* eta *doako solverrak*. Hala ere, badira beste mota batzuk ere, hala nola kode irekiko solverrak eta helburu espezifikoak dituzten solverrak.

#### 5.1.1.1. Solver komertzialak

Solver komertzialak euskarri tekniko eta etengabeko eguneratzeak ematen dituzten enpresek garatutako eta mantendutako optimizazio-tresnak dira. Solver horiek asko erabiltzen dira industrian, eraginkortasun handia, sendotasuna eta eskala handiko arazoak kudeatzeko gaitasuna dutelako. Hauek dira solver komertzial ezagunenak batzuk:

- **Gurobi:** bere abiadura eta doitasunagatik ezaguna, asko erabiltzen da ikerketan eta aplikazio industrialetan problema linealak, koadratikoak eta programazio ez-linealekoak ebazteko [5].
- **CPLEX:** IBMk garatua, ezaguna da programazio linealeko, programazio oso mistoko eta beste problema konplexu batzuetako problemak ebazteko duen gaitasunagatik [6].
- **Xpress:** aparteko errendimendua ematen du problema handiak eta konplexuak optimizatzeko, bereziki logistika eta telekomunikazioak bezalako industrietan [7].

Solver horiek izaten dira gehien aukeratzen direnak kalitate handiko soluzioa eta euskarri tekniko espezializatua behar direnean. Hala ere, horien kostua handia izan daiteke proiektu txiki edo akademikoetarako.

#### 5.1.1.2. Doako solverrak

Doako solverrak kosturik gabe erabiltzeko erabilgarri dauden optimizazio-tresnak dira. Askotan, solver komertzialen euskarri teknikorik eta ezaugarri aurreraturik ez badute ere, oso eraginkorrak izan daitezke eskala txikiagoko arazoetarako edo ikerketa-inguruneetarako. Hona hemen adibide batzuk:

- **GLPK (GNU Linear Programming Kit):** ezin hobea programazio lineal eta oso arazoetarako, aplikazio akademikoetan eta kode irekiko proiektuetan asko erabiltzen da [8].

- **COIN-OR (Computational Infrastructure for Operations Research):** kode irekiko proiektu bat, hainbat optimizazio-arazotarako solver-bilduma bat eskaintzen duena [9].
- **SciPy Optimize:** Python-eko SciPy liburutegiaren zati bat, optimizazio algoritmoen bariedade bat eskaintzen du arazo sinple eta nahiko konplexuetarako [10].

Doako solverra aukeratzea egokia izan daiteke solver komertzial baten gaitasun aurreratuak behar ez dituzten proiektuetarako edo aurrekontu-murrizketak dituzten proiektuetarako.

### 5.1.1.3. Minimize

Python-en minimize funtzioa da proiektu honetarako aukeratutakoa. SciPy liburutegiaren zati bat, doako solver-en taldekoa da. Funtzio jarraituak optimizatzeko problemak ebazteko tresna aproposa da. Minimize bereziki erabilgarria da funtzio ez-linealak inpliketzen dituzten optimizazioetarako, eta optimizazio-estrategia doitzeko metodo ugari eskaintzen ditu.

Proiektu honetarako minimize aukeratzea haren erabilerraztasunean, malgutasunean eta eskuragarri dagoen dokumentazio zabalean oinarritzen da, eta horrek Python aplikazioetan integratzea errazten du. Gainera, irtenbide eraginkorra da solver komertzial baten potentzia behar ez duten optimizazio-arazoetarako.

Minimize-en alternatibak daude Python ekosistemaren barruan, hala nola liburutegi espezializatuak erabiltzea *cvxpy* gisa optimizazio konbexurako, edo *Pyomo* erabiltzea modelatze matematiko aurreratuagorako. Hala ere, minimize aukeratu da bere sinpletasunagatik eta proiektuaren eskakizunekiko bateragarritasunagatik [11].

## 5.1.2. Reinforcement Learning

Reinforcement Learning ikaskuntza automatikoko teknika bat da, eragile batek ingurune batekin egindako interakzioaren bidez erabakiak hartzen ikasten duelako. Helburua da denboran zehar metatutako soluzio bat maximizatzea. VNF placement delakoaren

testuinguruan, Reinforcement Learning-ek aukera ematen du denboran eta erabileran hobetu daitezkeen konponbide egokitzaille optimoak aurkitzeko.

Reinforcement Learningen potentziala ingurune aldakorretara egokitzeko eta optimizazio-teknika tradizionalen bidez agerikoak ez diren estrategiak ikasteko duen gaitasunean datza. Hala ere, inplementazioa konplexuagoa izan daiteke, eta denbora esanguratsua behar izan daiteke modeloa entrenatzeko.

Reinforcement Learning-aren abantailak:

- Sarearen topologian denbora errealeko aldaketetara egokitzeko gaitasuna.
- Optimizazio-metodo tradizionalak baino irtenbide berritzaileagoak aurkitzeko aukera.
- Kostu operatiboen murrizketa potentziala, sarearen eraginkortasuna denborarekin hobetzen baita.

Reinforcement Learning-aren desabantailak:

- Konplexutasun handiagoa ereduaren inplementazioan eta entrenamenduan.
- Benetan entrenatzeko datu kopuru handiak behar izatea.
- Entrenamendua ondo diseinatuta ez badago, soluzio suboptimoen konbergentzia-arriskua.

Bere desabantailak gorabehera, Reinforcement Learning-a epe luzerako etorkizun handiko irtenbidea da, batez ere adimen artifizialaren gaitasunek eboluzionatzen jarraitzen duten heinean [12].

## 5.2. Erabiltzailearen interfazea garatzeko tresnak: Python, Tkinter

Proiektu honen erabiltzaile-interfazea garatzeko, Python programazio-lengoaia aukeratu da Tkinter liburutegiarekin batera. Python interpretatutako, goi-mailako eta helburu orokorreko programazio-lengoaia da, irakurgarritasunagatik eta sinpletasunagatik

ezaguna. Tkinter Python-en liburutegi estandar bat da, erabiltzaile-interfaze grafikoak (GUI) modu erraz eta eraginkorren sortzeko erabiltzen dena.

### 5.2.1. Zergatik Python eta Tkinter:

Python kode irekiko lengoaia bat da, eta, beraz, irisgarria eta malgua da maila guztietako garatzaileentzat. Industrian asko erabiltzen da, liburutegi eta frameworks bilduma zabala duelako, eta, horri esker, askotariko arazoei eraginkortasunez aurre egin dakieke. Gainera, Pythonek garatzaile komunitate handia du, eta horrek arazoak konpontzea eta baliabideak lortzea errazten du [13].

Tkinter Python hedapen bat da, interfaze grafiko sinpleak baina funtzionalak sortzea errazten duena. Erabiltzeko erraztasuna eta Python-ekin duen berezko integrazioa direla eta, aukera ezin hobea da oinarritzko GUI bat behar duten proiektuetarako, ezaugarri konplexuen beharrik gabe. Tkinter praktikoa da eta funtsezko elementuak ematen ditu erabiltzaile-interfaze intuitiboak eta funtzionalak garatzeko [14].

### 5.2.2. Garapen-tresnen beste aukera batzuk

Python eta Tkinter izan ziren proiektu honetarako aukeratutako tresnak, baina funtsezkoa da onartzea beste aukera asko daudela erabiltzaile-interfazeak garatzeko, bakoitzak bere abantaila eta erabilera-kasu espezifikoekin, hala nola: JavaScript, HTML, eta CSS; Java eta JavaFX; C #eta WPF; etab.

## 5.3. Optimizazio-metodoen aukerak

Minimize funtzioaren barruan, optimizazio-metodo desberdinen artean hauta daiteke, bakoitza ezaugarri bakarrekin eta erabilera-kasu espezifikoekin. Proiektu honetan, trust-constr metodoa aukeratu da murrizketak maneiatzeko duen gaitasunagatik eta optimizazio-arazo ez-lineal konplexuetan duen eraginkortasunagatik. Hala ere, hainbat hautabiderek probak egin dira, VNF placement arazoarekiko egokitasuna ebaluatzeko. Jarraian, alternatiba horiek eta horien eta trust-constr metodoaren arteko funtsezko desberdintasunak deskribatzen dira.

### 5.3.1. Trust-Constr

Trust-constr metodoa konfiantzazko eskualdeetan oinarritutako optimizazio-algoritmo bat da, arazoak murrizketekin konpontzeko erabiltzen dena. Metodo horrek dinamikoki doitzen du bilaketa-eremua, aurreko hurbilketen kalitatearen arabera, eta bereziki eraginkorra da arazoetarako, non murrizketek berebiziko garrantzia duten konponbide optimoan.

Hona hemen trust-constr sistemaren abantailak:

- **Murrizketen erabilera eraginkorra:** optimizazioan berdintasun-murrizketak eta desberdintasun-murrizketak sartzea ahalbidetzen du, eta hori funtsezkoa da VNF placement arazoa behar bezala modelatzeko.
- **Egonkortasuna eta Konbergentzia:** problema konplexuetan egonkortasuna eta konbergentzia probabilitatea hobetzen duen ikuspegi sendoa eskaintzen du.
- **Malgutasuna:** ondo egokitzen da funtzio objektibo mota desberdinetara, leunak ez direnak edo etenak dituztenak barne.

Trust-constr-ek berezko murrizketak erabiltzen dituenez, proiektu honetarako hautaketa egokia da, non sare-murrizketak eta VNF eskakizunak kontu handiz erabaki behar diren [15].

### 5.3.2. Optimizazio-metodoen beste aukera batzuk

Garapenean zehar, trust-constr metodoaren alternatiba batzuk aztertu ziren minimize funtzioaren barruan, eta jarraian zehazten dira:

#### L-BFGS-B

L-BFGS-B metodoa BFGS aldaera bat da, eta memoria-hurbilketa mugatu bat erabiltzen du funtzioak kaxa-murrizketekin optimizatzeko (bound constraints). Ezaguna da memorian eraginkortasuna eskatzen duten problemetan, eta murrizketa sinpleak aplikatzen zaizkie aldagaiei.



Trust-constr ez bezala, L-BFGS-Bk ez du berdintasun edo desberdintasuneko murrizketa orokorrik erabiltzen, eta horrek murrizketa horiek funtsezkoak diren arazoetara mugatzen du aplikagarritasuna.

Eraginkorragoa da memoriaren erabileran trust-constr baino, baina malgutasun txikiagoa du murrizketak erabiltzean [16].

### **Nelder-Mead**

Nelder-Mead metodoa sinplexean oinarritutako optimizazio algoritmoa da, deribatuak behar ez dituen. Funtzio etenak edo deribaezinak optimizatzeko baliagarria da, eta bere sinpletasunagatik eta implementatzeko erraztasunagatik da ezaguna.

Nelder-Meadek ez du murrizketarik erabiltzen, eta, beraz, ez da hain egokia sare-murrizketak kritikoak diren arazoetarako.

Ez du deribatuen informaziorik behar, eta hori abantaila izan daiteke egoera jakin batzuetan, baina eraginkortasuna mugatzen du deribatuek optimizaziorako informazio baliotsua ematen duten problemetan [17].

### **TNC**

TNC gradiente konjokatua kutxa-murrizketekin optimizazio-problema ebazteko erabiltzen duen metodoa da. Egokia da problema handietarako eta murrizketa sinpleak dituztenentzako, zehaztasunaren eta eraginkortasun konputazionalaren arteko balantzea eskainiz.

Aurreko biek bezala, TNCK kaxa-murrizketak baino ez ditu maneiatzen eta ez da gai murrizketa orokorrak maneiatzeko, eta horrek murrizketa konplexuagoak dituzten arazoetara mugatzen du bere aplikagarritasuna.

Eskala handiko arazoetan eraginkorragoa da, gradiente bateratuetan duen ikuspegia dela eta, baina proiektu honetarako beharrezkoa den malgutasuna sakrifikatzen du.

Hasieran, minimize funtzioa metodo bat zehaztu gabe exekutatu zen, eta, horren ondorioz, BFGS metodo lehenetsia erabili zen. Hala ere, konbergentzia-arazoak aurkitu

ziren optimizazio-probleman dauden murrizketak behar bezala maneiatzeko BFGSren ezintasunarekin lotuta. Faktore hori erabakigarria izan zen trust-constr aukera egiteko, murrizketen erabilera sendoagoa ematen baitu eta VNF placement arazorako irtenbide egonkorra bermatzen baitu [18].

## 6. Proposatutako irtenbidearen deskribapena

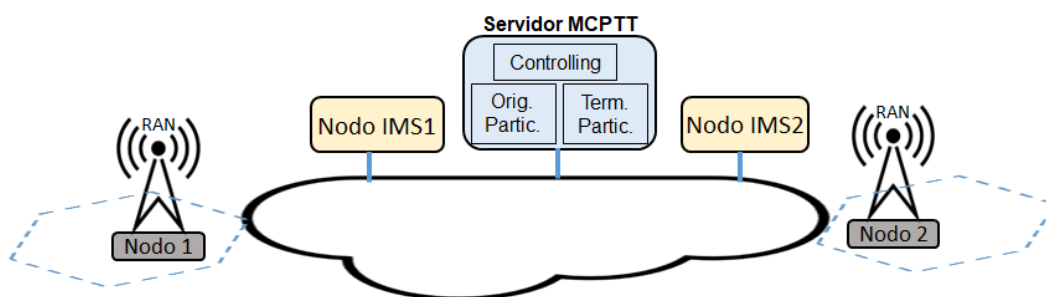
Txostenaren atal honetan, 5G sareetan VNF funtzioen kokapena optimizatzeko proposatutako irtenbidea deskribatzen da, Misio Kritikoko zerbitzuen eraginkortasuna eta errendimendua hobetzeko helburuarekin. Bi alderdi nagusi jorratuko dira: eredu matematikoa, sare-topologia, trafiko motak, VNF funtzioak, etab. barne hartzen dituena; eta eredu ebazpenerako software aplikazioa, backend, frontend, datu-basearen garapenean erabilitako teknologiak, analitik lortutako emaitzak, etab. zehazten dituena. Atal horiek ikuspegi osoa ematen dute eredu teorikoa eta proiektuaren helburuak lortzeko inplementazio praktikoa nola integratzen diren ikusteko.

### 6.1. Eredu matematikoa

Azpiatal honetan, ereduaren ezaugarri espezifikoak eta osagaiak zehaztuko dira hurrengo ataletan, eta haren funtzionamendua eta 5G eta Beyond 5G sareen testuinguruan dituen onurak sakonki ulertuko dira, misio kritikoko zerbitzuetarako [3].

#### 6.1.1. Sare topologia

*Irudia 4: bi nododun sare topologia.*



Misio kritikoko zerbitzu horietarako aintzat hartutako arkitekturak funtsezko osagai hauek ditu:

- Misio Kritikoaren aplikazio-zerbitzaria (MCX Server): zerbitzari honek funtzio kritikoen kudeaketa eta eragiketa zentralizatzen ditu, koordinazioa eta erantzun eraginkorra bermatuz.
- Sareko operadorearen IMS nodoa: MC sistemarekin seinaleztapen-mezuak trukatzear arduratzen da. Nodo horrek sareko zati desberdinen arteko integrazioa eta komunikazio arina ziurtatzen ditu.
- SARBIDE-nodoen multzoa  $\{N1, N2, N3...\}$ : baliabide konputazionalak hornituta daude, eta nodo horiek VNF funtzioak exekutatzeke gai dira, baliabideen banaketa malgua eta eskalagarria ahalbidetuz.
- Erabiltzaileen multzoa  $\{u1, u2, u3...\}$ : erabiltzaileak sarera konektatzen dira dagozkien sarbide-nodoen bidez, eta sistemak kudeatu behar duen eskari dinamikoa eta heterogeneoa adierazten dute.

Horrez gain, proposatutako ereduak baliabideak kudeatzeko mekanismo aurreratuak eta optimizazio-algoritmoak ditu, VNFen hedapena denbora errealean egokitzeko, lan-kargan eta sareko baldintzetan izandako aldaketei erantzunez. Ikuspegi integral horrek eraginkortasun operatiboa hobetzeaz gain, egoera kritikoen aurrean sarearen erresilientzia eta erantzuteko gaitasuna indartzen ditu.

## 6.1.2. Trafiko klaseak

Proposatutako ereduan, bost trafiko-mota nagusi definitzen dira, bakoitza Misio Kritikoko (MC) zerbitzuak emateko beharrezkoak diren komunikazio-motei dagokiena:

- **Seinaleztapen-trafikoa (T1):** trafiko mota honek Misio Kritikoaren zerbitzuak kudeatzeko eta kontrolatzeko behar diren komunikazioak barne hartzen ditu. Zerbitzu horiek emateko, SIP sare-nukleo bat behar da, hala nola IP multimedia-azpisistema (IMS). Operadore mugikorrek dagoeneko zabaldua dute IMS sistema lan-esparru nagusi gisa. MC sistemaren eta IMS nukleoaren arteko komunikazioa batez ere kontrol-eragiketetan gertatzen da, hala nola deiak eta saioak ezarri, mantendu eta amaitzean. Trafiko mota hori funtsezkoa da konexioak konfiguratzeko eta kudeatzeko, Misio Kritikoko zerbitzuen erabiltzaileak beti modu fidagarrian konektatuta egongo direla ziurtatuz.

- **Zoru-kontrolako trafikoa (T2):** trafiko-mota hau MCPTT (Push To Talk Misió Kritikoa) sistemetan komunikazio-kanalera sartzeko kontrolari dagokio. MCPTT komunikazio batean, funtsezkoa da une bakoitzean hitz egiteko eskubidea nork duen kudeatzea, "pisu-kontrola" izenez ezagutzen dena. Trafiko horrek ingurunerako sarbidea erabakitzeke trukaturako kontrol-mezuak barne hartzen ditu, erabiltzaile bakar batek batera hitz egitea eta hiztunen arteko trantsizioa arina eta gatazkarik gabea izatea ziurtatuz. Kontrol hori funtsezkoa da talde-komunikazioen argitasuna eta eraginkortasuna mantentzeko, batez ere koordinazioa eta erantzun-denbora kritikoa diren larrialdi-egoeretan.
- **PTT trafikoa (Push to Talk) (T3):** klase honek MCPTT zerbitzuaren erabiltzaileen ahots-trafikoa hartzen du. PTT trafikoa ahots-datuen fluxua da, eta erabiltzaile batek "hitz egiteko" botoia sakatzen duenean eta taldeko gainerako erabiltzaileei transmititzen zaienean sortzen da. Trafiko hori modu eraginkorrean kudeatu behar da, komunikazioak argiak eta atzerapen nabarmenik gabekoak izango direla bermatzeko, eta hori funtsezkoa da misio kritikoko agertokietan. Ahots-zerbitzuaren kalitatea, latentzia txikia eta fidagarritasuna funtsezkoak dira trafiko mota horretarako.

Hiru trafiko mota horiez gain, beste bi mota garrantzitsu daude Misio Kritikoen komunikazioetan: **MCDATA trafikoa (T4)** eta **MCVIDEO trafikoa (T5)**. MCDATA trafikoak testu-mezuak eta GPS lokalizazioa bezalako datuak barne hartzen ditu; MCVIDEO trafikoa, berriz, denbora errealeko bideo-transmisioei dagokie. Trafiko-mota horiek ezaugarri bereizgarriak dituzten arren, PTT trafikoaren antzeko portaera dute baliabideei eta latentziari dagokienez. Horregatik, lan honetan egindako VNF placement-aren ebaluazioan, aipaturako lehen hiru trafiko-motetan bakarrik jarri da arreta.

Bost trafiko-mota horiek Misio Kritikoko sare batean kudeatu eta optimizatu behar diren datuen eta seinaleen fluxuak adierazten dituzte. Mota horietako bakoitza behar bezala lehenestea eta maneiatzea funtsezkoa da Misio Kritikoko zerbitzuek modu eraginkorrean eta fidagarrian funtziona dezaten bermatzeko.

### 6.1.3. VNF funtzioak

Proposatutako ereduak onartzen du posible dela misio kritikoko zerbitzua birtualizatzea, haren funtzionalitateak bereiziz VNF desberdinetan, eta horiek sareko hainbat puntutan zabal daitezke errendimendua optimizatzeko. T1, T2 eta T3 klaseak bakarrik kontuan hartuta, VNF funtzioak honela definitzen dira:

- **VNF-1:** funtzio honek IMS seinaleztapen-trafikoa prozesatzen du (T1 mota). Gaur egun, prozesamendu hori sareko operadorearen IMS sisteman eta Misio Kritikoaren zerbitzarian (MCX Server) egiten da. Proposamen berriarekin, prozesamendu hori IMS nodoan egiten jarraituko da, baina, aitzitik, ez da nahitaezkoa MCX zerbitzarian exekutatzeko. Horregatik, VNF-1IMS (IMS nodoan exekutatzen dena) eta VNF-1 (MCX zerbitzarian edo erabilgarri dagoen sareko beste nodoren batean exekuta daitekeena) bereiziko ditugu.
- **VNF-2:** sarbide-kontrolako trafikoa (zoru-kontrola) prozesatzen duten funtzioak (T2 mota). Funtzio horiek funtsezkoak dira Push-To-Talk (PTT) komunikazioetako hizketa-txanda kudeatzeko.
- **VNF-3:** Push-To-Talk (T3 mota) ahots-trafikoa prozesatzen duen funtzioa. Hori da larrialdiko komunikazio-zerbitzuen funtzio kritikoenetako bat, ahots-komunikazioen transmisio eraginkorra eta azkarra bermatzen baitu.
- **VNF-S** eta **VNF-D:** sare birtualizatuko ingurune batean, MCX zerbitzuek azpiegitura fisikoa partekatzen dute beste zerbitzu batzuekin. Beraz, sarbide-nodoetan, irrati-interfazea eta baliabide konputazionalak "slice" desberdinen artean partekatuta daude. VNF-Sek uplink (igoerako) trafiko osoaren kontsumo konputazionala adierazten du, nodoak jasotakoa; VNF-Dk, berriz, nodotik konektatutako erabiltzaileei bidalitako downlink (jaitsierako) trafiko osoaren kontsumo konputazionala. Sarbide-nodo bakoitzak gutxienez VNF-S bat eta VNF-D bat izango ditu, VNF-S1/VNF-D1 gisa identifikatuak N1 sarbide-nodorako, VNF-S2/VNF-D2 N2 sarbide-nodorako, eta horrela hurrenez hurren.

Zerbitzuaren birtualizazioarekin, litekeena da VNF horiek sareko hainbat puntutan hedatzea, errendimendua hobetzeko eta latentzia murrizteko. Funtzio batzuek, hala nola VNF-1IMS eta sarbide-nodoetako VNF-S eta VNF-D funtzioek, kokapen finkoak dituzte

hasieratik. Hala ere, beste funtzio batzuetarako, hala nola VNF-1, VNF-2 eta VNF-3, kokapen optimoa bilatuko da, sistemaren baldintzetan eta ezarritako errendimendu-irizpideetan oinarrituta.

Inplementazio tradizionaletan, funtzio horiek guztiak MCX zerbitzarian bizi ohi dira, baina proposatutako ereduak aukera ematen du funtzio horiek sarean zehar modu malguagoan eta eraginkorragoan banatzeko, eta horrek nabarmen hobe ditzake sistemaren latentzia eta eraginkortasuna.

Eredu berriak B5G sare baten gaineko Misio Kritikoko zerbitzu-multzo baten hornikuntzan inplikaturako baliabide konputazionalak eta biltegiatze-baliabideak irudikatzen ditu. Helburua da zehaztea nola hedatu behar den VNF funtzio multzo bat ezarritako errendimendu-adierazleak (KPIak) betetzeko. Geroago azalduko den bezala, VNF funtzio bakoitza M/M/1 ilara baten bidez irudikatzen da; ilara-teorian sinplifikazio komun bat da, eta zerbitzu-baliabide bakarra duen zerbitzari batean eskaeren prozesamendua modelatzeko aukera ematen du. Horrek itxaronaldia eta sistemaren karga aztertzea errazten du.

#### 6.1.4. Taldeak

Misio kritikoko sareetan, taldeek funtsezko zeregina dute komunikazioen kudeaketan. Taldeak elkarren artean aldi berean komunikatu daitezkeen erabiltzaile-multzo batek osatzen ditu, eta, horrela, larrialdi-egoeretan edo lankidetz-egoeretan informazio kritikoa koordinatzea eta transmititzea errazten da.

Kontzeptu hori argitzeko, erabiltzaile-taldeen definizio hau hartu behar da kontuan: demagun  $G_i$  taldeak definituta daudela horietako bakoitzari dagozkion  $u_j$  erabiltzaileekin. Adibidez,  $G_1 = \{u_1, u_2, u_3, u_4\}$  eta  $G_2 = \{u_1, u_5, u_6\}$  zenbakiak adierazten dute  $G_1$  izeneko taldea  $u_1, u_2, u_3$  eta  $u_4$  erabiltzaileek osatzen dutela, eta  $G_2$  izeneko taldea, berriz,  $u_1, u_5$  eta  $u_6$  erabiltzaileek. Adibidean ikus daitekeenez, erabiltzaile bat talde bat baino gehiagoko kidea izan daiteke.

Taldeek ahalbidetzen dute erabiltzaile batek bidalitako Push-To-Talk (PTT) trafikoko ahots-mezu bat automatikoki banatzea taldeko kide guztiei. Mekanismo hori funtsezkoa

da Misio Kritikoko eragiketetarako; izan ere, informazio garrantzitsua berehala zabaltzen da kide guztien artean, koordinazioa eta erantzuna denbora errealean optimizatuz.

### 6.1.5. Eredua hasierako datuak

Eredua xehetasunez definitu eta optimizazio-arazoari heldu aurretik, funtsezkoa da hasierako datu multzo bat izatea, ingurunea eta baldintza operatiboak behar bezala ezaugarritzeko. Datu horiek honako hauek dira:

- **Erabiltzaileen kokapena:** funtsezkoa da erabiltzaile bakoitza sarera konektatuta dagoen sarbide-nodoaren bidez jakitea. Informazio horri esker, trafikoaren banaketa geografikoa zehaztu eta nodo bakoitzaren gaineko karga ebaluatu daiteke.
- **Erabiltzaile bakoitzak sortutako trafiko-kantitatea:** trafiko-mota (T1, T2 edo T3) eta trafiko horren xedea zehaztuta. Ahots-trafikoaren kasuan (PTT), mezuaren hartzaila espezifikoa identifikatzen da.

Hasierako datu horiekin, VNF bakoitzak erabili behar duen trafikoa kalkula daiteke. Gainera, VNFetako bat asetzetik gertu dagoen edo dagoeneko gainezka dagoen ebalua daiteke, eta horrek VNF horren instantzia gehigarriak sortzeko beharra adieraziko luke. Garrantzitsua da nabarmentzea, saturazio-egoerarik aurreikusten ez bada ere, VNFen erreplikak edo instantziak sor daitezkeela sistemaren sendotasuna eta eraginkortasuna hobetzeko.

### 6.1.6. Ilara-sare eredua

Misio kritikoko zerbitzuen atzerapenak optimizatzeko proposatutako eredua M/M/1 ilara-sare batean oinarritzen da, non  $Q$  multzoko ilara bakoitza,  $q$ , VNF funtzioa den. Ilara horien bezeroak VNFak prozesatzeko eskaerak dira. Trafiko mota bakoitza (T1, T2, T3)  $K$  multzoko  $k$  bezero-mota bat da [19].

#### 6.1.6.1. Ilaren ezaugarriak

$q$  ilara bakoitzak  $\mu_q$  zerbitzu-tasa bat du, dagokion VNFaren errendimendua islatzen duena. Zerbitzu-tasa hori zuzenean lotuta dago VNFren batez besteko exekuzio-denborarekin,  $\bar{t}_{vnf}$  bezala adierazita, non  $\bar{t}_{vnf} = 1/\mu_q$ . Garrantzitsua da azpimarratzea



VNFk CPUa okupatzeko behar duen denbora gauzatze-denbora baino luzeagoa izan daitekeela, protokoloen funtzionamenduari lotutako itzarote-denborak izan baitaitezke, besteak beste.

VNF bakoitzak kontsumo konputazional bat ere badu  $v_q$  parametroaren bidez, beharrezko baliabide konputazionalen kopurua adierazten duena. Gainera, VNF bakoitzari lotutako energia-kontsumoa hartzen da kontuan,  $w_q$  parametroaren bidez adierazia.

### 6.1.6.2. Trafikoaren iritsiera- eta banaketa-tasa

$K$  mota bakoitzerako  $q$  ilara bakoitzera iristen diren eskaeren tasa, berriz,  $\lambda_k^q$ .  $q$  ilara batera iritsitako tasa totala que gisa irudikatzen da, hau da, klaseko iritsiera-tasa guztien batura:

$$\lambda_q = \sum_{k \in K} \lambda_k^q$$

### 6.1.6.3. Sareko nodoak

Sareko nodoak  $N$  multzoak adierazten ditu, eta multzo horretan sartzen dira IMS nodoa, misio kritikoko zerbitzaria (MCX) eta sarbide-nodo batzuk (N1, N2, N3...).  $n \in N$  nodo bakoitzak  $V_n$  ahalmen konputazionala du bertan instalatutako VNFetarako. Gainera, nodo bakoitzak  $W_n$  energia-kontsumo baimenduaren gehieneko kantitatea du instalatutako VNFen multzorako.

### 6.1.6.4. Loturak eta latentziak

Nodoak sare-loturen bidez konektatuta daude, eta lotura zuzenak edo birtualak izan daitezke. Bi  $n_l$  eta  $n_m$  nodoen artean lotura fisiko zuzena badago, loturaren ahalmena  $C_{l,m}$  gisa adierazten da. Hainbat loturaren bidezko konexioen kasuan, lotura birtual bakartzat hartzen dira, eta loturaren ahalmena gaitasun indibidual txikiagoa duen loturari dagokio. Bi  $n_l$  eta  $n_m$  nodoen arteko sare-latentzia  $\delta_{l,m}$  bezala adieraz daiteke.

### 6.1.6.5. Eskarien transferentzia

$P(q_2|q_1, k)$  motako transferentzia-probabilitate espezifikoek adierazten dute  $k$  motako zerbitzu-eskaera bat  $q_2$  VNFra sartzeko probabilitatea,  $q_1$  VNFk erantzun ondoren.  $k$

motako eskaera bat VNF  $q$ -n prozesatzen hasteko probabilitatea  $P(q, k)$  gisa adierazten da.

## 6.1.7. Optimizazio-problema

Proiektu honen helburua VNF funtzioen hedapen optimoa zehaztea da, zerbitzuaren atzerapena minimizatuz, hori baita gure errendimendu-metrika nagusia. Hori lortzeko, helburu funtzio bat erabiltzen da, atzerapena murrizketa batzuen pean minimizatzeko [19].

### 6.1.7.1. Erabaki aldagaiak

Erabaki-aldagai nagusi bat dago eta honela definitzen da:

- $A(h, q) \in \{0,1\}$ : VNF  $q$   $h$  hostean kokatuta dagoen adierazten duen aldagai bitarra.  $A(h, q) = 1$  hedatzen bada, eta  $A(h, q) = 0$  bestela.

### 6.1.7.2. Sistemaren murrizketak

1. **Host-en konputazio gaitasuna:** host bakoitzaren konputazio gaitasuna ezin da gainditu:

$$\sum_{q \in Q} A(h, q) \mu(q) \leq \kappa_h, \quad \forall h \in H.$$

2. **Sistemaren egonkortasuna:** VNFek nahikoa gaitasun konputazional izan behar dute beren karga maneiatzeko:

$$\lambda_k(q) = \lambda_{k,q} + \sum_{p \in Q} P(q|p, k) \lambda_k(p).$$

$$\Lambda(q) = \sum_{k \in K} \lambda_k(q).$$

$$\Lambda(q) < \mu(q), \quad \forall q \in Q.$$

3. **Lotura gaitasuna:** hosten arteko loturen edukiera ezin da gainditu:

$$\sum_{k \in K} \sum_{q, r \in Q} \lambda_k(q) P(r|q, k) A(h, q) A(l, r) \leq C(h, l).$$

4. **Energia-kontsumoa:** ez da gainditu behar nodo bakoitzean baimendutako gehieneko energia-kontsumoa:

$$\sum_{q \in Q} A(h, q)w_q \leq W_h, \quad \forall h \in H.$$

### 6.1.7.3. Latentzia eta erantzun-denborak

Zerbitzu-atzerapena VNFetako prozesamendu-denboren eta hainbat nodoren arteko sare-latentzien konbinazio bat da. M/M/1 ilarak suposatuz,  $k$  zerbitzu-eskaera bat VNF  $q$ -n prozesatzeko denbora hau da:

$$R_k(q) = \frac{1}{\mu(q) - \Lambda(q)}, \quad \forall q \in Q.$$

$k$  zerbitzu-motaren eskaerak espero duten sare-latentzia hau da:

$$\sum_{q,r \in Q} \gamma_k(q)P(r|q, k) \sum_{h,l \in H} \delta(h, l)A(h, q)A(l, r).$$

$k$  zerbitzu generikoaren eskaeren batez besteko guztizko atzerapena honela gertatzen da:

$$D_k = \sum_{q \in Q} \gamma_k(q)R_k(q) + \sum_{q,r \in Q, q \neq r} \gamma_k(q)P(r|q, k) \sum_{h,l \in H} A(h, q)A(l, r)\delta(h, l).$$

### 6.1.7.4. Helburu funtzioa

Helburu funtzioa  $k$  zerbitzu-mota bakoitzaren gehieneko atzerapen normalizatua minimizatzeko formulatzen da, QoS-ren mugarekiko:

$$\min_{A, \mu} \max_{k \in K} \frac{D_k}{D_{QoS, k}}.$$

### 6.1.7.5. Ebazteko metodoa

Optimizazio-problema ebazteko, urrats hauek egingo dira:

1. **Arazoaren bertsio ganbila planteatzea:**  $A(h, q)$  aldagai bitarren ordeztan  $\tilde{A}(h, q) \in [0,1]$  aldagai jarraituak jartzea. Aldagaien produktuak maneiatzeko,  $\Phi(l, q, m, r)$  beste aldagai bat sartu.

$$\Phi(l, q, m, r) \leq \tilde{A}(l, q),$$

$$\Phi(l, q, m, r) \leq \tilde{A}(m, r),$$

$$\Phi(l, q, m, r) \geq \tilde{A}(l, q) + \tilde{A}(m, r) - 1.$$

2. **Araza konbexua solver batekin ebaztea:** emandako murrizketen arabera planteatutako arazoa konponduko duen solverra erabiltzea.
3. **Heuristika bat aplikatzea VNFak esleitzeko:** Z-score erabiltzea hedapenerako ilara eta nodo optimoak zehazteko.

$$Z(h, q) = \tilde{A}(h, q) + 1(A(h, q)\mu(q) \geq T_{h,q}).$$

$$(h^*, q^*) \leftarrow \arg \max_{h \in H, q \in Q} Z(h, q).$$

Errepikatu urrats horiek, VNF guztien hedapena zehaztu arte.

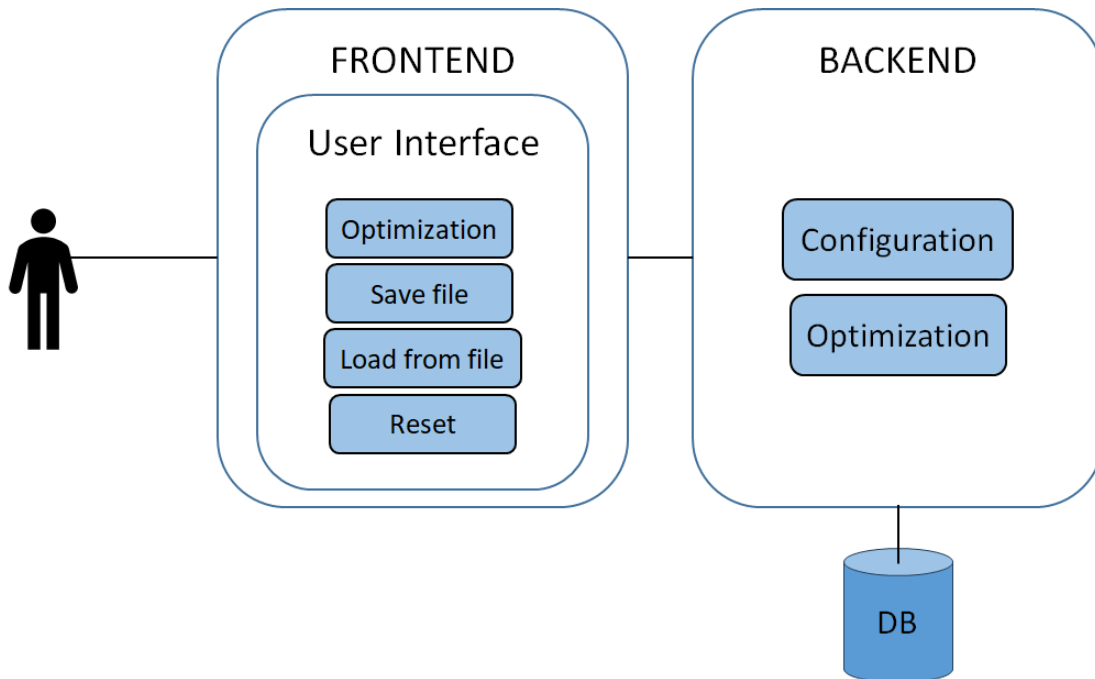
## 6.2. Eredu ebazpenerako software aplikazioa

Azpiatal honetan azaldu berri den eredu matematikoaren ebazpena lortzen duen softwarearen diseinua aurkeztuko da.

### 6.2.1. Softwarearen arkitektura nagusia

Softwarearen arkitektura hiru osagai nagusik osatzen dute: frontend, backend eta datu basea (ikus 5. irudia). Osagai horietako bakoitzak rol espezifiko bat du, eta elkarrekin komunikatzen dira funtzionaltasun osoa eta eraginkorra emateko. Hala ere, frontend-a da gainerako osagai guztiekiko interakzioa hartzen duen elementu bakarra. Jarraian, osagai horiek eta dagozkien funtzioak zehazten dira.

*Irudia 5: softwarearen arkitektura nagusia.*



Frontend erabiltzaile-interfazea da, eta aplikazioarekin elkarrean zuzena izatea ahalbidetzen du. Interfaze honen bidez, erabiltzaileek sistemaren eragiketarako funtsezkoak diren hainbat ekintza egin ditzakete. Frontend-aren funtzionalitate nagusiak hauek dira:

- **Optimization:** VNF funtzioen kokapena optimizatzeko prozesua hasteko aukera ematen dio erabiltzaileari. Prozesu hori funtsezkoa da sareko baliabideak modu eraginkorrean erabiliko direla ziurtatzeko.
- **Save:** gorde gaur egungo konfigurazioa eta optimizazioaren emaitzak datu-basean. Hori baliagarria da konfigurazioen eta emaitza historikoen erregistroa mantentzeko.
- **Load from file:** fitxategi batetik konfigurazioak eta datuak kargatzen ditu. Horri esker, erabiltzaileak aurreko konfigurazioak berrerabil ditzake, datu guztiak berriro sartu behar izan gabe.
- **Reset:** konfigurazio eta datu guztiak bere hasierako egoerara itzultzen ditu. Funtzio hau erabilgarria da berriro hasteko, aurretiko konfiguraziorik gabe.

Frontend-a datuak gordetzeko eta berreskuratzeko datu basearekin komunikatzen da, eta backend-arekin prozesatzeko eta optimizatzeko lanak egiteko.

Backend sistemaren muina da, eta bertan egiten dira prozesatzeko eta optimizatzeko eragiketa nagusiak. Osagai hau kalkulu intentsiboak eta datuen kudeaketa konplexua eskatzen dituzten funtzioak exekutatzeari arduratzen da. Backend-aren funtzionalitate nagusiak hauek dira:

- **Configuration:** sistemaren hasierako konfigurazioak kudeatzen ditu. Horrek barne hartzen ditu hasierako datuak kargatzea eta ingurunea optimizazio-prozesurako prestatzea.
- **Optimization:** optimizazio-prozesua egiten du algoritmo aurreratuak erabiliz, VNF funtzioen kokapen onena zehazteko. Prozesu hau konfigurazio-datuetan eta erabiltzaileak definitutako aldagaietan oinarritzen da.

Backend-a frontend-arekin komunikatzen da erabiltzailearen komandoak eta datuak jasotzeko, eta datu-basearekin optimizazio-eragiketarako beharrezkoa den informazioa bildu eta berreskuratzeko.

Datu-basea sistemaren informazio garrantzitsu guztia biltegitzeaz arduratzen den osagaia da, konfigurazioak, optimizazio-emaitzak eta softwarearen etengabeko funtzionamendurako beharrezkoa den beste edozein informazio barne. Datu-baseak aukera ematen du datuak etengabe biltegitzeko, informazioa eraginkortasunez berreskuratzeko, eta konfigurazioen eta optimizazio-emaitzen historiala mantentzeko.

## 6.2.2. Erabilitako teknologiak

Aplikazioa garatzeko, hainbat teknologia, lengoia eta liburutegi erabili dira, backend-ari eta frontend-ari laguntzen diotenak, konponbide integrala eta eraginkorra emanez. Jarraian, erabilitako teknologia nagusiak deskribatzen dira:

### 6.2.2.1. Python programazio lengoia

Aplikazio hau garatzeko erabiltzen den lengoia nagusia da. Python bere sinpletasun eta irakurgarritasunagatik da ezaguna, eta horrek kodearen garapena eta mantentzea errazten

du. Bere liburutegi-bilduma zabalak eta euskarri komunitarioak magnitude desberdinetako proiektuetarako ezin hobea egiten dute.

**Tkinter:** Python hedapen bat da, erabiltzaile-interfaze grafikoak (GUI) modu erraz eta eraginkorrean sortzeko aukera ematen duena. Tkinter Python-en liburutegi estandarren parte da, eta horrek instalazio gehigarrien beharra ezabatzen du eta mahaigaineko aplikazioen diseinua errazten du [13].

**NumPy:** liburutegi hau funtsezkoa da Python-en zenbakizko konputaziorako. Konponketak eta matrize multidimentsionalak egiteko euskarria ematen du, baita datu horiekin jarduteko goi-mailako funtzio matematikoen bilduma zabala ere. Aplikazio honetan, NumPy optimizazio-prozesuan beharrezkoak diren kalkuluak eta datu-egiturak maneiatzeko erabiltzen da [13].

**SciPy:** NumPy osatzen duen liburutegia da eta matematika aurreratuko, zientziako eta ingeniartzako lanetarako erabiltzen da. SciPyk optimizaziorako moduluak, aljebra lineala, integrazioa, interpolazioa eta beste eragiketa matematiko konplexu batzuk biltzen ditu. Proiektu honetan, SciPy erabiltzen da optimizazio-algoritmoan beharrezko kalkulu aurreratuak egiteko [13].

**Time:** Time liburutegia programaren zati desberdinen exekuzio-denbora neurtzeko erabiltzen da, eta horrek errendimenduan botila-lepoak optimizatzen eta identifikatzen laguntzen du [13].

**PIL** (Python Imaging Library): zehazki, PILen Image klasea erabiltzen da aplikazioaren barruan irudiak maneiatu eta manipulatzeko. Horren barruan irudiak kargatzea, aldatzea eta bistaratzea sar daitezke [13].

**Os:** liburutegi honek sistema eragilearekin elkarreragiteko modu egokia eskaintzen du. Hainbat zereginetarako erabiltzen da, hala nola fitxategiak eta direktorioak manipulatzeko, sistemaren komandoak exekutatzeko eta sistemaren inguruneari buruzko informazioa lortzeko [13].

**Subprocess:** subprocess liburutegiak aukera ematen du prozesu berriak sortzeko, sarrera/irteera/errore hodietara konektatzeko eta itzulera-kodeak lortzeko. Aplikazio

honetan, optimizazio-prozesuan beharrezkoak diren kanpoko script-ak exekutatzeke subproces-ak erabiltzen dira [13].

### 6.2.2.2. MariaDB datu-basea

MariaDB erabili da optimizazioen emaitzak biltegitratzeko eta kudeatzeko datu-basearen plataforma gisa. MariaDB MySQLren adarkatze bat da, bere bateragarritasun, errendimendu eta sendotasun handiagatik ezaguna, eta horrek aukera fidagarri bihurtzen du datuen erabilera eraginkorra eskatzen duten aplikazioetarako.

### 6.2.3. Backend

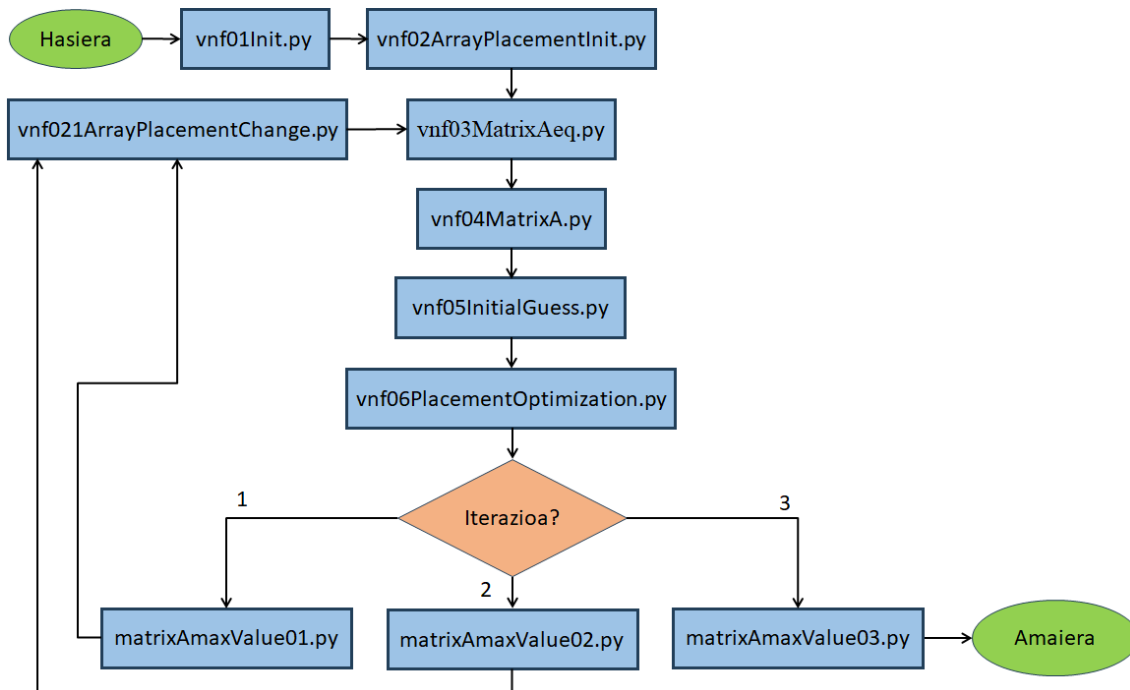
Aplikazio honen backend-a funtsezkoa da optimizazio-prozesua gauzatzeko, eta sekuentzia espezifiko batean script batzuk exekutatzeke diseinatuta dago. Script nagusiak, fullSequence izenekoak, aldagaiak abiarazteke, kalkuluak egiteke eta VNF funtzioen posizionamendua optimizatzeke beharrezkoak diren beste script batzuk exekutatzen ditu.

#### 6.2.3.1. Moduluaren egitura

Backendaren programazioa antolatzeke, 6. irudian ikus daitekeen hainbat azpimodulu inportatzen dira: *vnf01Init*, *vnf02ArrayPlacementInit*, *vnf03MatrixAeq*, *vnf04MatrixA*, *vnf05InitialGuess*, *vnf06PlacementOptimization*, *matrixAmaxValue01*, *vnf021ArrayPlacementChange*, *matrixAmaxValue02*, *matrixAmaxValue03*. Azpimodulu horiek optimizazio-prozesuaren urrats bakoitzerako beharrezko funtzioak dituzte.



**Irudia 6: moduluaren fluxu-diagrama.**



*Main* funtzioak tenporizadore bat hasten du guztizko konputazio-denbora neurtzeko, eta, gero, hasieratzearekin hasten da. Lehenik, *vnf01Init.main* eta *vnf02ArrayPlacementInit.main* exekutatzen dira beharrezko aldagaiak eta VNFen hasierako kokapena konfiguratzeko.

Ondoren, VNF-1, VNF-2 eta VNF-3 kontsideratzen direnez, hiru optimizazio-iterazio egiten dira. Lehen iterazioan, hainbat script exekutatzen dira sekuentzian (*vnf03MatrixAeq*, *vnf04MatrixA*, *vnf05InitialGuess*, *vnf06PlacementOptimization*, *matrixAmaxValue01*) hasierako datuak erabiliz kokapen-matrizea kalkulatzeko eta optimizatzeko. Etapa honetan, *vnf06PlacementOptimization*-etik abiatuta lehen iterazioaren *delay1* (atzerapen optimoaren lehen hurbilketa) balioa lortzen eta gordetzen da.

Bigarren iterazioan, lehenengo iterazioaren irteeran oinarritutako hasierako kokapena doitzen da, eta kalkuluak honako script hauekin errepikatzen dira: *vnf021ArrayPlacementChange*, *vnf03MatrixAeq*, *vnf04MatrixA*, *vnf05InitialGuess*,

*vnf06PlacementOptimization*, eta *matrixAmaxValue02*. Hemen, *delay2* (atzerapen optimoaren bigarren hurbilketa) balioa lortzen eta gordetzen da.

Hirugarren iterazioan, bigarren iterazioaren irteeran oinarritutako doikuntza- eta kalkulu prozesua errepikatzen da, *vnf021ArrayPlacementChange*, *vnf03MatrixAeq*, *vnf04MatrixA*, *vnf05InitialGuess*, *vnf06PlacementOptimization*, eta *matrixAmaxValue03* exekutatu. Etapa honetan, *delay3* (atzerapen optimoaren azken balioa) eta *placement* (bilatzen ari den VNF placement) arrayaren balioak lortzen eta gordetzen dira.

Iterazioak bete ondoren, zenbatzeko denbora osoa kalkulatu da eta azken kokapen optimizatu lortzen da (aurretiaz aipatutako *placement* arraya deritzona). Exekuzioan gertatzen den edozein salbuespen atzitu eta maneiatzen da, eta, beharrezkoa bada, errore-mezu bat itzultzen da. Scripta zuzenean exekutatzen bada, main funtzioari deitzen zaio, eta zenbaketa-denboraren emaitzak, azken kokapena eta delay balioak inprimatu dira.

### 6.2.3.2. Abiarazte-aldagaiak

Abiarazte-aldagaiak funtsezkoak dira optimizazio-programa honen backend eta frontende. Aldagai horiek *vnf01Init* scriptean kargatu dira eta optimizazio-prozesuak behar bezala gauzatzeko beharrezko parametroak ezartzen dituzte. Jarraian, aldagai horiek eta programaren testuinguruan duten funtzioa zehazten dira.

- ***numVNF***: aztertu beharreko sare-funtzio birtualen (VNFen) kopurua definitzen du. Kasu honetan, 3 balio dituen aldagai finko bat da, hiru trafikoko motatarako VNF funtzioen optimizazioan oinarritzen baita programa: seinaleztapena, zoru-kontrola eta ahotsa (PTT).
- ***numHosts***: VNF funtzio bakoitza koka daitekeen nodoen kopurua adierazten du. Bi nodo dituen sareko topologia batean, bere balioa 3 izango litzateke: 1. nodoa, 2. nodoa eta MCX zerbitzaria.
- ***deltaArray***: nodoen arteko sare-latentziak irudikatzen dituen matrize bat da. Latentziek hainbat faktoreren eragina izan dezakete: distantzia fisikoa, seinalearen hedapena, transmisio ingurunearen mota eta kalitatea, sare ekipamendua, kongestioa, sare protokoloak, bideratzeak, software konfigurazioak, interferentzia elektromagnetikoak eta firewalls bezalako segurtasun neurriak.

- ***kpi1, kpi2, kpi3***: aldagai hauek finkoak dira eta sistemak trafiko klase bakoitzeko iturritik helmugarako bete behar dituen denbora maximoak esleitzen ditu.
- ***muVNF***: erabiltzaileak finkatutako aldagai bat da, VNFen zerbitzu-tasa adierazten duena. Aldagai hau funtsezkoa da sare-funtzioen atzerapenak kalkulatzeko.
- ***pIturriHelmuga***: erabiltzaileak definitutako aldagaiak dira, eta sareko nodoetan dauden erabiltzaileen artean zirkulatzen duen ehunekoa adierazten dute. Adibidez, 2 nodoko topologian, 1. Nodoan (p11), 1 eta 2. nodoen artean (p12, p21) eta 2. nodoaren barruan (p22) dauden erabiltzaileen artekoa izango litzateke.
- ***rhoSIP, rhoVNF1, rhoVNF2, rhoVNF3, rhoNodo***: aldagai hauek nodoen batez besteko okupazioa adierazten dute, eta erabiltzaileak finkatzen ditu. Ilara/nodoetako batez besteko atzerapenak kalkulatzeko erabiltzen dira, hau da, erabiltzaile batek zerbitzatu aurretik itxaroten duen denbora kalkulatzeko.
- ***delaySx, delayDx, delaySIP, delayVNF1, delayVNF2, delayVNF3***: aldagai hauek aurrez zehaztutako aldagaieetatik abiatuta kalkulaten dira, eta ilara/nodoetako bataz besteko atzerapenak irudikatzen dituzte.
- ***delayVNFk1, delayVNFk2, delayVNFk3***: klase bakoitzeko trafikoa jatorrizko erabiltzailetik helmugako erabiltzaileraino osatzeko behar diren atzerapen-denborak irudikatzen dituzte.

Azken aldagai horiek hobeto ulertzeko, kontuan har dezagun trafiko mota bakoitza zerbitzu bat dela, eta horietako bakoitzerako zerbitzu-kate bat defini daiteke:

- **T1 klaseko trafikoa (seinaleztapena)**: zerbitzu honen fluxua sarbide-nodoen VNFs-Setan hasten da, IMS nodoaren VNF1 funtzioarekin jarraitzen du, VNFMCX funtziora pasatzen da eta honen erantzuna IMS sistemara itzultzen da, eta azkenik dagokion erabiltzaileari helarazten zaion nodoaren VNF-D funtzioaren bidez birbideratzen da. Hots, jatorria eta helmuga berdinak izan daitezke.

- **T2 motako trafikoa (zoru-kontrola):** zerbitzu honi dagokion fluxua sarrerako nodoetako edozein VNF-Stan hasten da, VNF2 funtzioarekin jarraitzen du eta helmugako nodoetako edozein VNF-Dtan amaitzen da.
- **T3 klaseko trafikoa (ahotsa - PTT):** trafiko honetarako zerbitzu katea T2 klaseko trafikoaren antzekoa da.

### 6.2.3.3. Funtzio nagusiak eta optimizazio prozesuak

Programaren backendean erabiltzen diren bi optimizazio-funtzio garrantzitsuenak *linprog* eta *minimize* dira. Funtzio horiek funtsezkoak dira aplikazioan planteatzen diren optimizazio-arazoak konpontzeko.

#### *linprog*

Linprog funtzioa SciPy liburutegiak programazio linealeko problemak konpontzeko ematen duen tresna bat da. Programazio linealeko arazo baten helburua da murriztapen lineal batzuei lotutako helburu funtzio lineal bat minimizatzea (edo maximizatzea). Arazoaren forma orokorra hau da:

Minimizatu:

$$\min_x c^T x$$

Subjektua:

$$A_{ub}x \leq b_{ub},$$

$$A_{eq}x = b_{eq},$$

$$l \leq x \leq u,$$

Programan hasierako soluzio bat aurkitzeko erabiltzen da eta honela adierazita dago:

*linprog* (*fInitialGuess*, *A\_ub=A*, *b\_ub=b*, *A\_eq=Aeq*, *b\_eq=beq*, *bounds=bounds*, *method='simplex'*)

non,

- $f_{InitialGuess}$ : minimizatu beharreko helburu funtzio linealaren koefizienteak dituen arrai bat da.
- $A_{ub} = A$ : desberdintasun-murrizketen matrizea da.  $A_{ub}$ -en lerro bakoitzak  $x$  desberdintasun-murrizketa lineal baten koefizienteak zehazten ditu,  $x$  erabaki aldagaien bektorea izanik.
- $b_{ub} = b$ : desberdintasun-murrizketen bektorea da. Elementu bakoitzak  $A_{ub}x$ -ri dagokion balioaren goiko muga bat adierazten du.
- $A_{eq} = Aeq$ : berdintasun-murrizketen matrizea da.  $A_{eq}$  lerro bakoitzak  $x$  motako berdintasun-murrizketa lineal baten koefizienteak zehazten ditu.
- $b_{eq} = beq$ : berdintasun-murrizketen bektorea da.  $A_{eq}x$  elementu bakoitzak  $b_{eq}$  elementu egokiaren berdina izan behar du.
- $bounds = bounds$ :  $x$ -ren elementu bakoitzerako pareen sekuentzia bat da (minimoa, maximoa), erabakitzekeo aldagai horren gutxieneko eta gehieneko balioak definitzen dituena.
- $method = 'simplex'$ : problema ebazteko erabilitako algoritmoa da. "Simplex" metodoa programazio linealerako algoritmo ezagunenetako bat da, eta soluzio egingarrien erpinetan zehar mugituz funtzionatzen du, soluzio optimoa aurkitzeko.

### ***minimize***

Minimize funtzioa optimizazio problema ez-linealak konpontzeko SciPyren tresna orokorra da. Murriztapenak dituen eskalatzeko-funtzio bat minimizatzeko aukera ematen du. Gure programan, problema lineal gisa modelatu ezin diren optimizazio-problema konplexuagoak ebazteko erabiltzen da.

Programan, minimize funtzioa honela erabiltzen da:

```
minimize (f6, x0, constraints=constraints, bounds=bounds, method='trust-constr')
```

non,

- $f6$ : minimizatu beharreko funtzio objektiboa da.  $fun(x, * args) \rightarrow float$  bezala definituta dago, non  $x$  ( $n$ ,) forma duen matrize dimentsiobakar bat den eta  $args$  funtzioa erabat zehazteko beharrezkoak diren parametro finkoen tupla bat den.
- $x0$ : hasierako estimazioa da. ( $n$ ,) tamainako elementu errealeen matrize bat da, non  $n$  aldagai independenteen kopurua den.
- *constraints*: murrizketen definizioa da. Murrizketa horiek hainbat motatakoak izan daitezke, hala nola berdintasun- edo desberdintasun-murrizketak.
- *bounds*: erabakitzeke aldagaien mugak dira. Bi eratara zehaztu daitezke: Bounds motako instantzia bat bezala edo  $x$  motako elementu bakoitzerako pareen sekuentzia bat bezala (minimoa, maximoa). Erabiltzen ez badira, ez da mugarik zehazten.
- *Method = 'trust-constr'*: erabilitako solver mota da. "Trust-constr" metodoa optimizazio algoritmo bat da, murriztapenen menpe dagoen eskalatze funtzio bat minimizatzen duena. Metodo hau bereziki erabilgarria da arazo ez-linealetarako, eta konfiantzazko eskualdean oinarritutako hurbilketa bat erabiltzen du konponbide optimoa aurkitzeko. Algoritmoak konfiantzazko eskualde bat egokitzen du momentuko soluzioaren inguruan, eta azpioptimizazio arazo bat konpontzen du eskualde horretan, soluzio optimorantz aurrera egiteko.

#### 6.2.4. Frontend

Gure proiektuaren frontend-ak funtsezko eginkizuna betetzen du erabiltzaile-interfaze grafiko intuitibo eta eskuragarria eskaintzen baitu sare-funtzio birtualizatuen (VNF) kokapenak optimizatzeko programarako. Modulu hau erabiltzailearen eta sistemaren arteko interakzioa errazteko diseinatuta dago, eta optimizazio-eragiketak modu erraz eta ulergarrian egiteko aukera ematen du. Interfazearen bidez, erabiltzaileek datuak sar ditzakete, optimizazio-prozesua exekutatu eta emaitzak argi eta eraginkortasunez bistaratu.

Lehenik eta behin, interfaze horren garapena gidatu zuten helburuez hitz egingo da, gero frontend-aren egitura zehazteko, nola antolatuta dagoen eta osagai nagusiak zein diren azalduz. Ondoren, horiek garatzeko erabiltzen diren teknologiak aztertuko dira. Azkenik, kodearen funtzio garrantzitsuenak deskribatuko dira, eta alderdi bakoitzak sistemaren funtzionamendu orokorrari nola laguntzen dion nabarmenduko da.

### 6.2.4.1. Moduluaren ezaugarriak

Frontend honen helburu nagusia mahaigaineko aplikazio simple eta funtzional bat garatzea da. Horretarako, tkinter erabiltzea erabaki da, Python-en integratutako liburutegi bat. Aplikazio honen helburua da adibide gisa adieraztea nola optimizazio-prozesu bat, hasiera batean zeregin espezifikotarako garatua, egokitu daitekeen erabiltzailearen interfaze grafiko (GUI) irisgarri bat emateko. Horrela, tresna teknikoaren erabilera demokratizatu nahi da, trebetasun tekniko desberdinak dituzten pertsonak sistemarekin modu eraginkorrean elkarreragin ahal izan dezaten.

Tkinter aukeratzeak Python-en berezko tresnak erabiltzeko beharrari erantzuten dio, interfazeak sortzea errazten dutenak, kanpoko mendekotasun konplexurik behar izan gabe. Horri esker, aplikazioa arina, banatzeko erraza eta epe luzera mantentzeko modukoa da. Interfaze grafiko aurreratuagoak dituzten beste liburutegi batzuekin alderatuta, tkinter nahiko oinarrizko irtenbidea bada ere, behar bezala betetzen ditu proiektu honen baldintzak, eta sinpletasuna eta erabiltzeko erraztasuna eskaintzen ditu.

Interfaze horren garapenaren helburu nagusietako bat erabiltzaile-esperientzia eskuragarria eskaintzea da. Interfazeak intuitiboa izan behar du, eta erabiltzaileei aukera eman behar die funtzionamendua azkar ulertzeko eta erabiltzeko, prestakuntza tekniko aurreratuaren beharrik gabe. Horrek esan nahi du UIen diseinu argia eta koherentea egin behar dela, nabigazioa eta berariazko zereginak gutxieneko esfortzu batekin egitea erraztuz.

Gainera, prozesu optimizatuekiko interakzioa erraztu nahi da. Interfazearen bidez, erabiltzaileek aurrez ezagutza tekniko espezializatua behar zuten optimizazio-prozesu konplexuekin elkarreragin dezakete. Horren ondorioz, GUItik eskura daitezkeen funtzionaltasun batzuk daude, eta algoritmo optimizatuak erraz eta zuzenean exekutatzeko aukera ematen dute.

Konponbidea moldakorra izateko eta askotariko testuinguru eta beharretara egokitzeko ere diseinatuta dago. Hasiera batean, optimizazio-lanen multzo espezifikotarako batean zentratzen bada ere, aplikazioaren arkitektura behar bezain malgua da pertsonalizazioa eta beste eremu batzuetara egokitzea ahalbidetzeko. Ikuspegi horrek aplikazioa hainbat eremutan erabiltzea errazten du, aukera eta aplikazio potentzial berriei atak irekiz.

## 6.2.4.2. Moduluaren egitura

Interfazearen egitura bi zatitan banatzen da: leiho nagusia eta X nodoko topologiaren leihoa.

### 6.2.4.2.1. *Leiho nagusia*

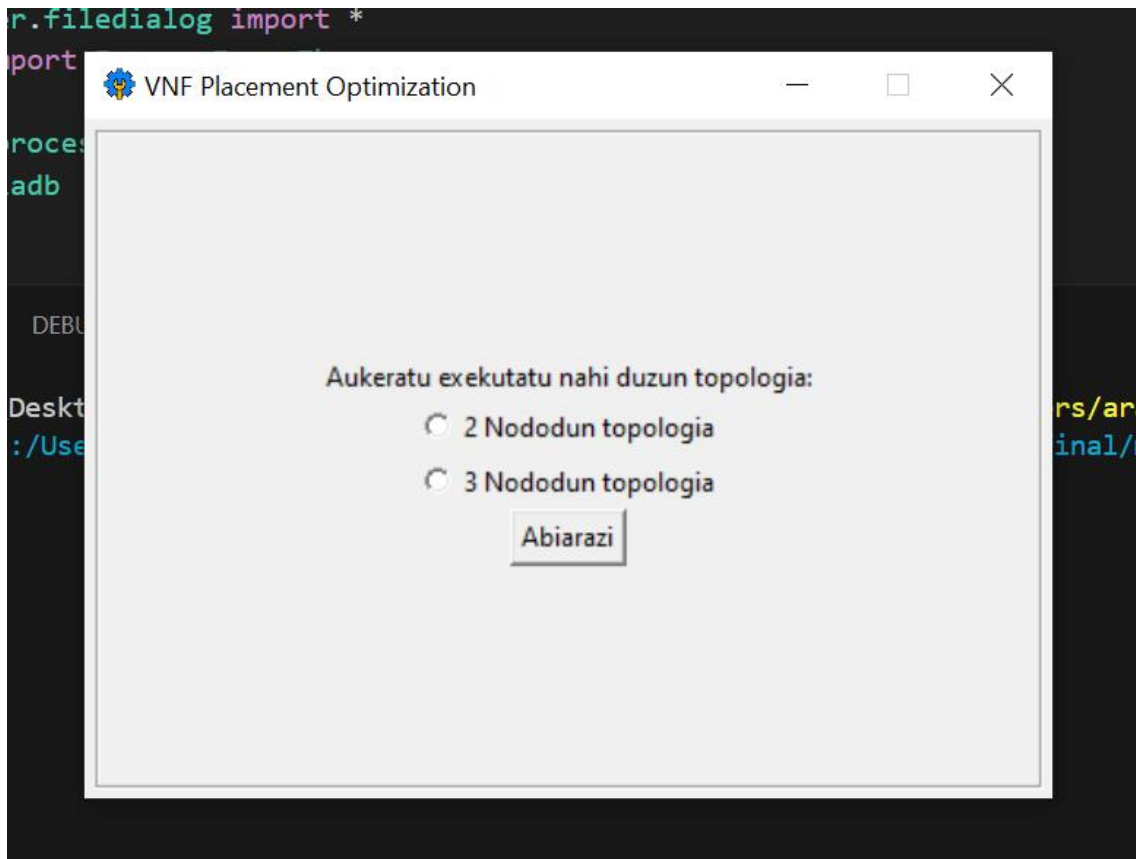
Programa leiho nagusiarekin hasten da (root), eta leiho horrek aukera soil bat ematen dio erabiltzaileari, zer topologia mota exekutatu nahi duen jakiteko. Bi aukera eskaintzen dira: 2 nodoko topologia eta 3 nodokoa.

Leiho nagusia tamainaz mugatuta dago, Tkinterren bi propietate garrantzitsu erabiliz: *geometry* eta *resizable*. Propietate geometrikoak leihoaren hasierako tamaina ezartzen du karaktere-kate batean 'widthxheight' formatuarekin, non zabalera eta altuera pixeletan neurtzen diren. Kasu honetan, topologia-aukeretarako espazio egokia eskaintzeko konfiguratu da, beharrezko tamaina gainditu gabe. Bestalde, *resizable* propietateak leihoaren tamaina aldatzeko ahalmena murrizten du. Implementazio horretan, (0,0) gisa konfiguratuta dago, eta horrek esan nahi du leihoa ezin dela berregokitu, ez zabalera, ez altueran, aurkezpen uniformeaa ziurtatuz, exekuzio-ingurunea edozein dela ere.

Ikusizko aurkezpena hobetzeko, *LabelFrame* widget-a erabiltzen da, testu-lerro bakoitza bisualki bereizitako marko baten barruan kokatu eta lerrokatzen duena. Horrek, itxura hobetzeaz gain, edukia logikoki eta modu eskuragarrian antolatzen du. Topologiaren hautaketa kudeatzeko, *Radiobutton* motako bi widget erabiltzen dira. Aukera-botoi horiek aukera ematen diote erabiltzaileari bi sare-konfigurazio posibleen artean intuitiboki hautatzeko, interakzio erraza eta zuzena eskainiz. *Radiobuttonen* aukeraketa bereziki egokia da hemen, erabiltzaileak aukera bakarra aldi berean hauta dezakeela bermatzen baitute, topologiaren aukeraketan anbigutasun posible oro ezabatuz.



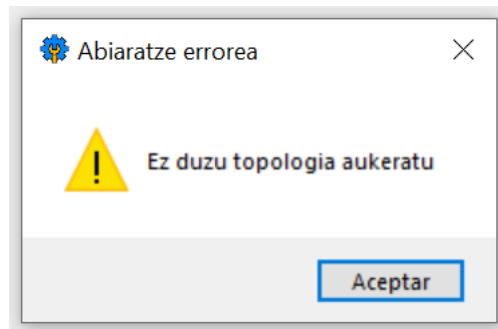
### Irudia 7: leiho nagusia.



Erabiltzailea leiho nagusian dagoenean, 7. irudian ikus daitekeen bezala, 2 nodoko topologiako leihora *Button* widget baten bidez sartzen da, “Abiarazi” izenaz deitua. Botoi hori erabiltzaileak *Radiobutton*en bidez erabilgarri dauden topologietako bat hautatu ondoren gaitzen da. Garrantzitsua da nabarmentzea *Radiobutton*ak, lehenetsita, hautatu gabe datozela, eta horrek esan nahi du erabiltzaileak hautaketa aktiboa egin behar duela jardun aurretik.

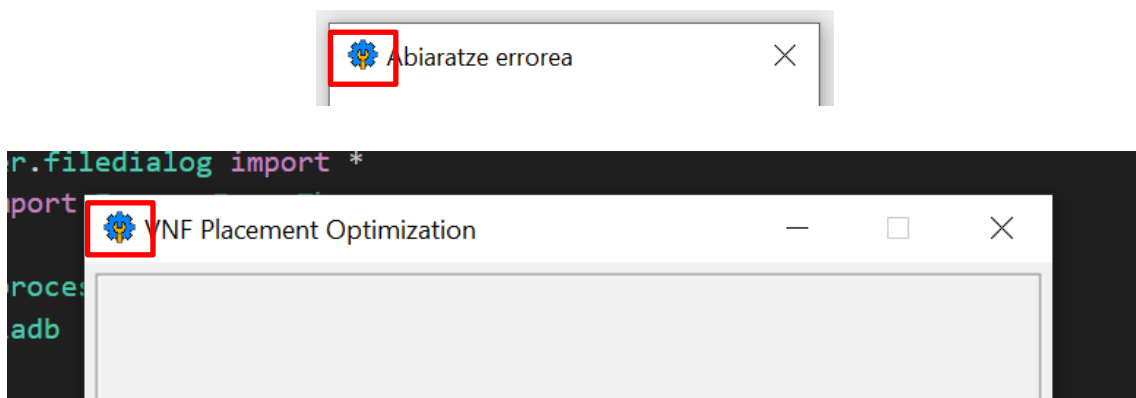
Erabiltzailea topologia batean sartzen saia daitekeen egoerak maneiatzeko, aurrez aukera bat hautatu gabe, kodea diseinatuta dago erroreen kontrol bat sartzeko *try-except* egitura baten bidez. Mekanismo horrek ziurtatzen du, topologietako bat aukeratu aurretik botoia sakatuz gero, programak ez duela hutsik egingo. Horren ordez, *messagebox* bat irekiko da, eta erabiltzaileari ohartaraziko zaio aukeretako bat hautatu behar duela jarraitu aurretik.

**Irudia 8: errorea mezua, "Ez duzu topologia aukeratu".**



Gainera, aplikazioaren pertsonalizazio bisualari arreta jarri zaio. Leiho guztietan, bai *messagebox* delakoetan, bai leiho nagusietan, beste ikono bat txertatu da, defektuzkoa ez bezalakoa. Aldaketa hori bisuala baino ez bada ere, aplikazioari nortasuna eta bereizketa ematen laguntzen du, beste interfaze estandarretatik bereiziz eta softwarearen identitate bisuala hobetuz.

**Irudia 9: ikono irudia.**

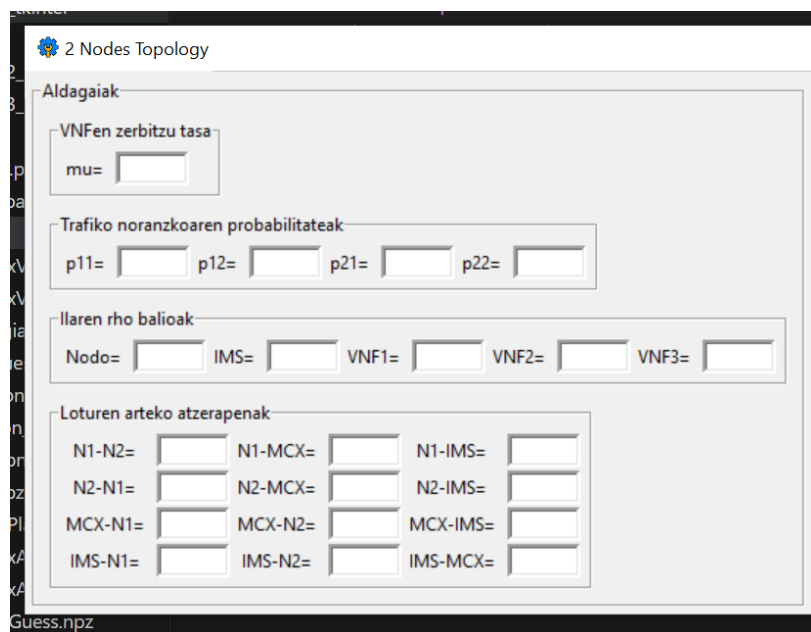


**6.2.4.2.2. Bi nodoko topologiaren leihoa**

Bi nodoko topologiaren leihoa aplikazioaren nukleo funtzionala da, eta programaren ezaugarri nagusi guztiak biltzen ditu, aurrerago xehetasunez azalduko direnak. Bisualki, leiho hori bi sekzio handitan banatuta dago: "ezkerreko hemisferioa" eta "eskuineko hemisferioa". Sekzio horiek horizontalki banatuta daude, erabiltzailearen interakzioa errazteko.

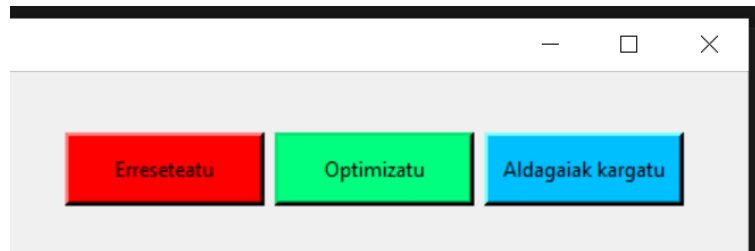
Ezkerreko hemisferioan sartzen ditu erabiltzaileak optimizazioa gauzatzeko beharrezkoak diren hasierako aldagaiak. Aldagai horiek hainbat azpitaldetan antolatuta daude, *LabelFrame* widget-a erabiliz, eta horrek aukera ematen du kategoria bakoitza argi bereizteko eta ulertzeko. Kategoriak honako hauek dira: VNFen zerbitzu-tasa, trafikoa noranzkoaren probabilitateak, nodo bakoitzaren rho balioak, eta nodo edo konexioen arteko atzerapenak. *LabelFrame* bakoitza "Aldagaiak" izeneko esparru handiago baten barruan dago, egitura koherentea eskaintzen duena eta nabigazioa errazten duena. Aldagai bakoitzerako, widget *Entry* erabili da datuak sartu ahal izateko eta widget *Label*, berriz, eremu bakoitza argi eta modu deskribatzailean etiketatzeko.

**Irudia 10: "Aldagaiak" *LabelFrame*-a.**



Bestalde, eskuineko hemisferioa hiru bloke nagusitan banatuta dago: bloke funtzionala, emaitzen blokea eta irudia. Bloke funtzionalean zeregin desberdinak betetzen dituzten hiru botoi daude: *Reset* botoiak aldagai guztien hasierako balioak berrezartzen ditu, *Optimizatu* botoiak optimizazio-prozesua erabiltzaileak sartutako aldagaien arabera gauzatzen du, eta *Aldagaiak kargatu* botoiak aukera ematen dio erabiltzaileari aurretik gordetako aldagaien konfigurazioak eta optimizazio-emaitzak inportatzeko.

### *Irudia 11: bloke funtzionala.*



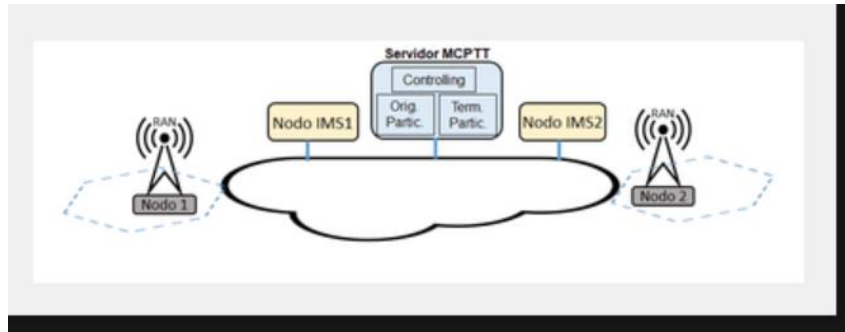
Emaitzen blokea optimizazio-prozesuaren irteera erakusteko izendatutako eremua da. Hemen azaltzen dira VNF funtzio bakoitzaren esleipena erabilgarri dauden nodoei, VNF bakoitzari lotutako atzerapena iturri-nodotik helmuga-nodora, eta gailuak optimizazio osoa egiteko behar duen konputazio-denbora. Hasieran, multzo hori ezkutuan dago, ez baitago emaitzarik eskuragarri optimizazio bat egin arte edo aurretik gordetako datuak kargatu arte. Emaitzak erakutsi ondoren, botoi gehigarri bat agertuko da, “Gorde” izenarekin etiketatua, momentuko aldagaiak eta optimizazioaren emaitzak gordetzeko.

### *Irudia 12: emaitzen blokea.*



Azkenik, irudi-bloke argigarri bat dago, bi nodoko sare baten irudikapen bisuala ematen duena, IMS nodoekin eta MCX zerbitzariarekin. Irudi horrek ikusizko laguntza gisa jokatzen du, eta aurkeztutako emaitzak testuinguruan kokatzen ditu, erabiltzaileari sare optimizatuaren egitura hobeto ulertzen lagunduz.

**Irudia 13: irudiaren blokea.**



**Irudia 14: bi nodoko topologiaren leihoa osotasunean.**

The screenshot shows a software interface titled '2 Nodes Topology'. It features several configuration sections and a results panel.

**Aldagaiak (Parameters):**

- VNFen zerbitzu tasa:** mu = 2000
- Trafiko noranzkoaren probabilitateak:** p11 = 0.25, p12 = 0.25, p21 = 0.25, p22 = 0.25
- Ilarren rho balioak:** Nodo = 0.1, IMS = 0.1, VNF1 = 0.1, VNF2 = 0.1, VNF3 = 0.1
- Loturen arteko atzerapenak (Delays):**

N1-N2 = 0.006	N1-MCX = 0.002	N1-IMS = 0.002
N2-N1 = 0.006	N2-MCX = 0.002	N2-IMS = 0.002
MCX-N1 = 0.002	MCX-N2 = 0.002	MCX-IMS = 0.001
IMS-N1 = 0.002	IMS-N2 = 0.002	IMS-MCX = 0.001

**Emaitza (Results):**

- VNF1: MCX
- VNF2: MCX
- VNF3: MCX
- Delay VNF1: 1.69 segundo.
- Delay VNF2: 1.42 segundo.
- Delay VNF3: 1.42 segundo.
- Konputazio denbora: 1.01 segundo.

Buttons include 'Erresetatu' (Reset), 'Optimizatu' (Optimize), 'Aldagaiak kargatu' (Load parameters), and 'Gorde' (Save). A small version of the network diagram from Figure 13 is shown at the bottom right.

### 6.2.4.3. Funtzio nagusiak

Atal honetan, bi nodoren topologia-leihoaren logika osatzen duten funtzio nagusiak deskribatzen dira: *reset\_n2*, *optimization\_n2*, *takefromfile\_n2*, eta *save\_n2*. Funtzio horietako bakoitzak berariazko zeregin bat betetzen du aplikazioaren barruan, eta datuen manipulazio egokia eta programaren fluxua ziurtatzen ditu.

#### Reset\_n2 funtzioa

*Reset\_n2* funtzioaren helburu nagusia hasierako egoerara sartzeko balioak berrezartzea da. Erabiltzaileak testu-eremuetan sartutako edozein datu garbitzeko, programa berriz exekutatzeko interfazea prestatzeko edo okerreko sarrerak zuzentzeko erabiltzen da.

Funtzioak *entries\_n2* izeneko hiztegi batean biltegiratutako sarrera-eremu guztiak zeharkatzen ditu. Eremu bakoitzerako, bertan dagoen edozein testu ezabatzen du, hutsik eta datu-sarrera berri baterako prest utziz.

Funtzio honek sarrerako eremuekin bakarrik elkarreragiten duenez, ez da beharrezkoa salbuespen konplexuak erabiltzea, ezabatze-eragiketek ez baitute errore kritikorik sortzen.

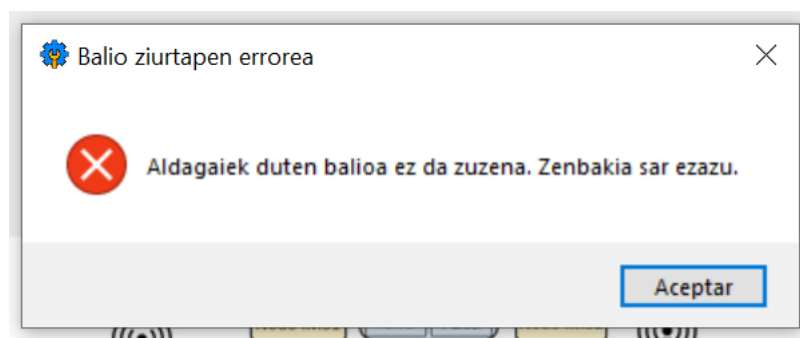
### Optimization\_n2 funtzioa

*Optimizazio\_n2* funtzioa optimizazio-prozesuaren muina da. Erabiltzaileak sartutako datuak atzematea, baliozkotzea eta sarearen topologia optimizatzeko beharrezko kalkuluak egiten dituen backend-ean ikusitako *fullSequence* script-a exekutatzea du helburu.

Funtzionamendu orokorra:

- **Datuak baliozkotzea:** hasieran, funtzioak sarrerako eremu guztiek balio numeriko baliodunak dituztela egiaztatzen du. Eremuren bat hutsik badago edo zenbakizkoak ez diren karaktereak baditu, salbuespena egingo du. (15. irudia)

#### *Irudia 15: balio ziurtapen errorea.*



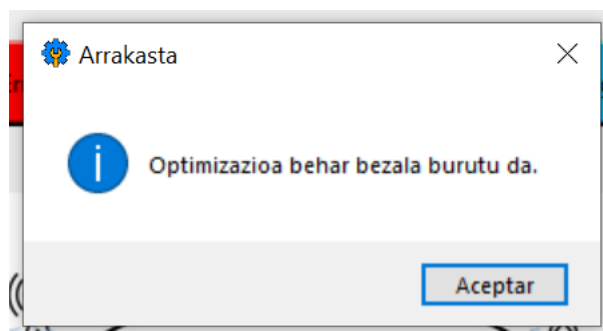
- **Balioak biltegiratzea:** baliozkotu ondoren, balioak testu-fitxategi batean biltegiratzen dira *nodo2\_values.txt* delakoan (16. irudia), eta erabiltzailearen sarreraren erregistro bat mantentzea ahalbidetzen dute.

***Irudia 16: aldagaiak temporalki gordetzen dituen fitxategia.***

```
node2_values.txt X
node2_values.txt
1 mu_str: 2000
2 p11_str: 0.25
3 p12_str: 0.25
4 p21_str: 0.25
5 p22_str: 0.25
6 rhoNodo_str: 0.1
7 rhoSIP_str: 0.1
8 rhoVNF1_str: 0.1
9 rhoVNF2_str: 0.1
10 rhoVNF3_str: 0.1
11 retN1N2_str: 0.006
12 retN1MCX_str: 0.002
13 retN1IMS_str: 0.002
14 retN2N1_str: 0.006
15 retN2MCX_str: 0.002
16 retN2IMS_str: 0.002
17 retMCXN1_str: 0.002
18 retMCXN2_str: 0.002
19 retMCXIMS_str: 0.001
20 retIMSN1_str: 0.002
21 retIMSN2_str: 0.002
22 retIMSMCX_str: 0.001
```

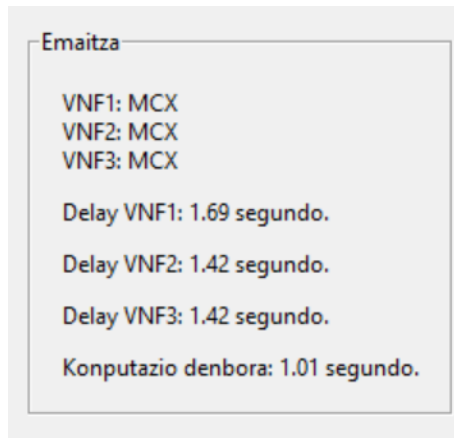
- **Script-a exekutatzea:** funtzioak topologia optimizatzen duen Python script bat (fullSequence.py) deitzen du, eta haren irteera jaso. Script-a ondo exekutatuz gero, informazio mezu bat helaraziko du. Script-ak errore-kode bat itzultzen badu, salbuespen baten bidez kudeatzen da. (17. irudia)

***Irudia 17: optimizazioaren arrakasta mezua.***



- **Emaitzen prozesamendua:** script-aren emaitzak atera eta prozesatzen dira, hala nola konputazio-denbora, VNF funtzioen kokapena eta bakoitzaren atzerapenak. Emaitza horiek formalizatu eta biltegitatu egiten dira, gero erabiltzeko edo bistaratzeko. (18. irudia)

### *Irudia 18: optimizazio emaitzak.*



Salbuespenen erabilera:

- **ValueError:** sarrera-balioak baliozkoak ez badira abiarazten da, erabiltzaileari errore-mezu bat erakutsiz, sarrerak zuzen ditzan.
- **RuntimeError:** kanpoko scripta ez bada behar bezala exekutatzen, mezu informatibo bat agertuko da, eta erabiltzaileak exekuzioan akats bat egon dela identifikatu ahal izango du.
- **General Exception Handling:** script exekutatzean ustekabeko edozein akats atzematen da, aplikazioak ustekabeen huts egiten ez duela bermatuz eta errore mezu lagungarria emanez.

### **Takefromfile\_n2 funtzioa**

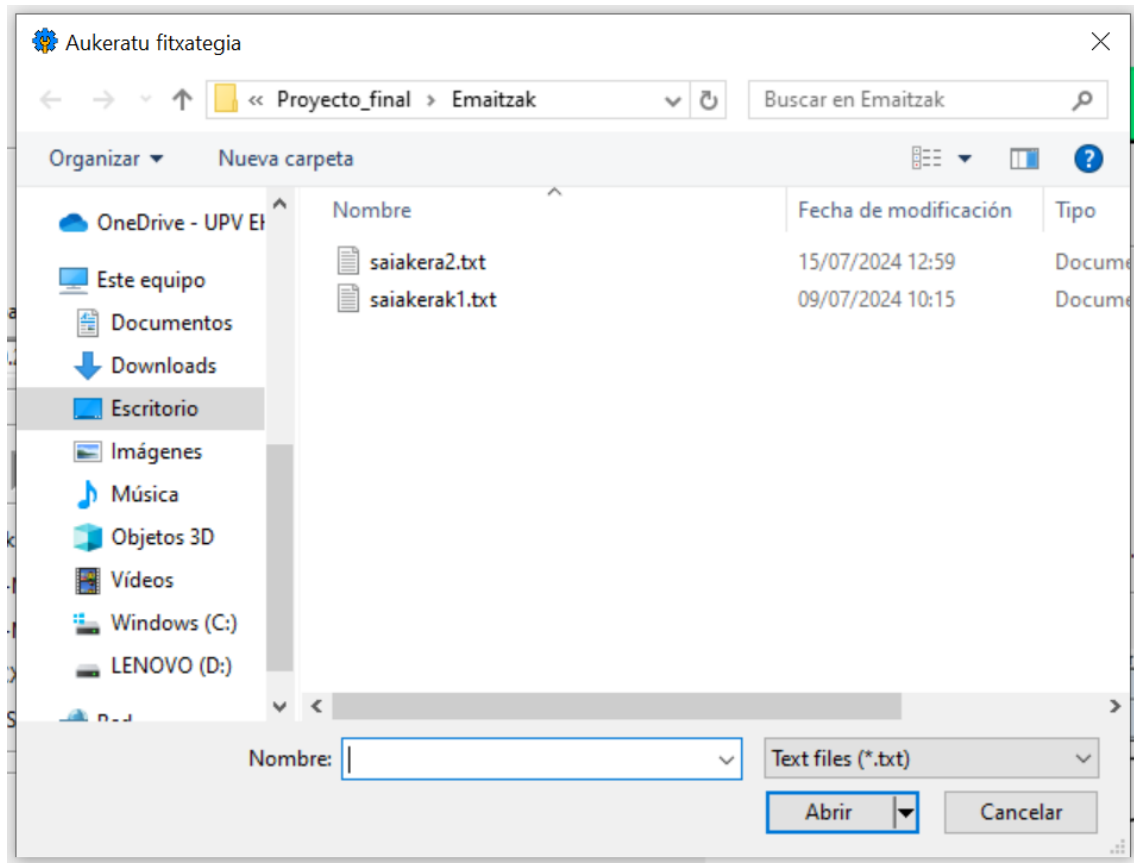
*Takefromfile\_n2* funtzioak alde aurretik gordetako fitxategi batetik balioak kargatzea errazten du, eta aurreko konfigurazioak berrerabiltzeko aukera ematen du, datu guztiak eskuz berriro sartu beharrik gabe.

Funtzionamendu orokorra:



- **Fitxategia hautatzea:** elkarrizketa-koadro bat irekitzen da, eta, horren bidez, erabiltzaileak testu-fitxategi bat hauta dezake datuak kargatzeko.

*Irudia 19: fitxategia aukeratzeko leihoa.*



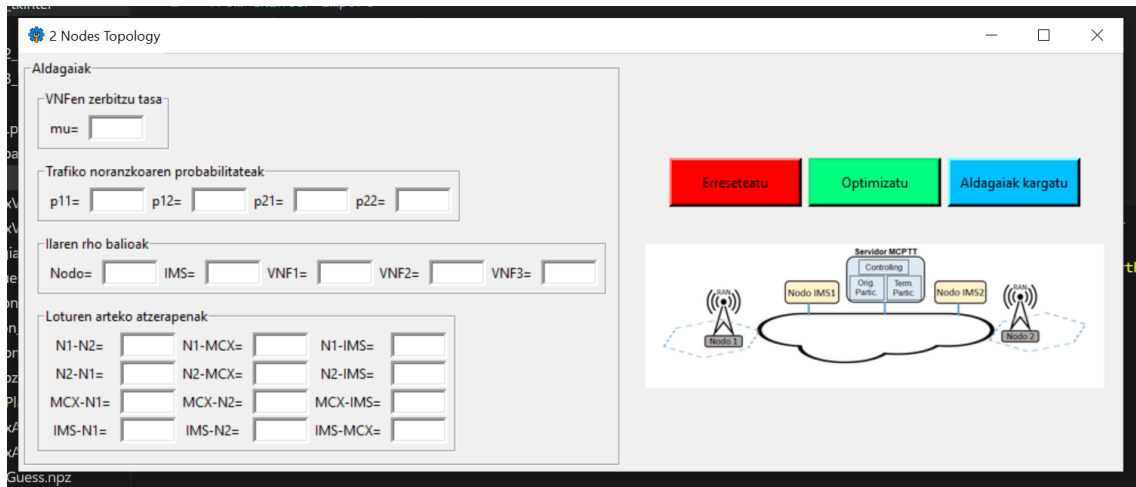
- **Datuak irakurtzea eta prozesatzea:** fitxategia hautatu ondoren, funtzioak edukia irakurri eta datuak ataletan sailkatzen ditu: konfigurazio-aldagaiak eta optimizazio-emaizak.

*Irudia 20: aurrez gordetako konfigurazioaren eta emaitzen fitxategia.*

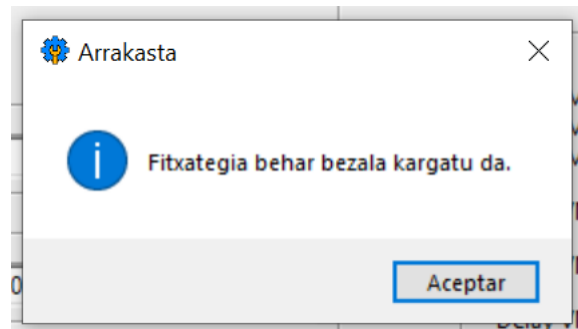
```
≡ saiakera2.txt ×
Emaitzak > ≡ saiakera2.txt
1 Aldagaiak:
2 mu_str: 2000
3 p11_str: 0.25
4 p12_str: 0.25
5 p21_str: 0.25
6 p22_str: 0.25
7 rhoNodo_str: 0.1
8 rhoSIP_str: 0.1
9 rhoVNF1_str: 0.1
10 rhoVNF2_str: 0.1
11 rhoVNF3_str: 0.1
12 retN1N2_str: 0.006
13 retN1MCX_str: 0.002
14 retN1IMS_str: 0.002
15 retN2N1_str: 0.006
16 retN2MCX_str: 0.002
17 retN2IMS_str: 0.002
18 retMCXN1_str: 0.002
19 retMCXN2_str: 0.002
20 retMCXIMS_str: 0.001
21 retIMSN1_str: 0.002
22 retIMSN2_str: 0.002
23 retIMSMCX_str: 0.001
24
25 Optimizazioaren emaitza:
26 VNF1: MCX
27 VNF2: MCX
28 VNF3: MCX
29 Konputazio denbora: 1.01 segundo
30 Delay VNF1: 1.69 segundo
31 Delay VNF2: 1.42 segundo
32 Delay VNF3: 1.42 segundo
```

- **Eremuak eguneratzea:** konfigurazio-aldagaiak dagozkien sarrera-eremuetan kargatzen dira. Optimizazio-emaitzak aldagai globaletan biltegitratzen dira, gero erabili edo erakusteko.

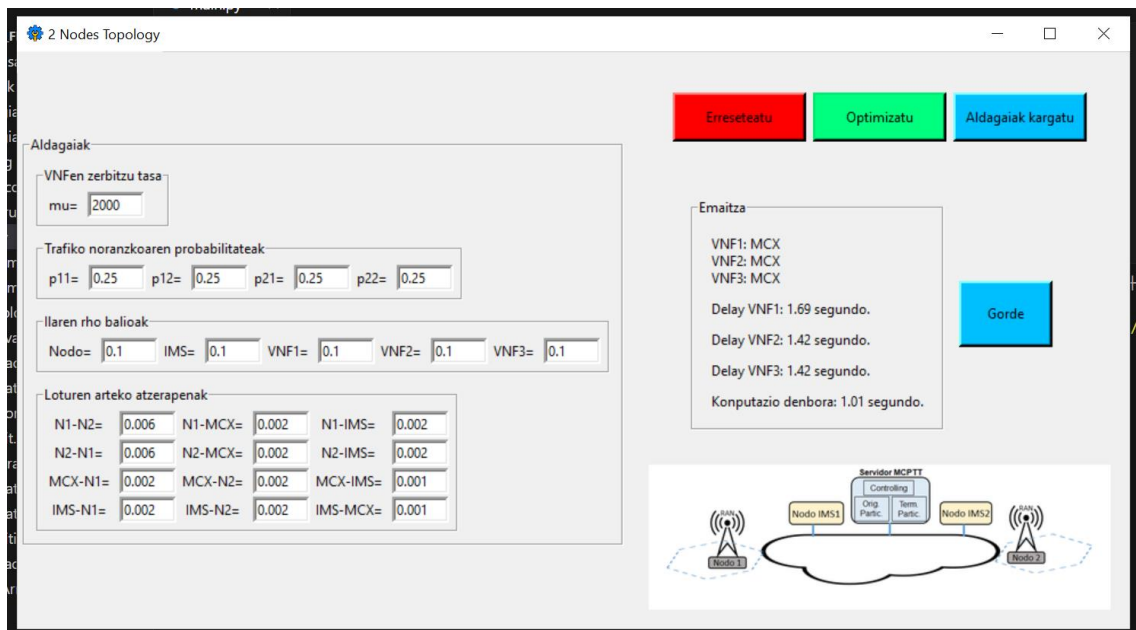
**Irudia 21: interfazea fitxategitik datuak kargatu orduko.**



**Irudia 22: fitxategia behar bezala kargatu denaren mezua.**



**Irudia 23: interfazea fitxategitik datuak kargatu ondoren.**



Salbuespenen erabilera:

- **FileNotFoundException** edo **IOError**: fitxategia irekitzearekin eta irakurtzearekin lotutako erroreak kudeatzen dira, eta kargatzean edozein arazoren berri ematen zaio erabiltzaileari.
- **General Exception Handling**: ustekabeko beste edozein salbuespen atzematen da, programak erabiltzaileari ustekabeko errore bat gertatzen bada informazioa emango diola ziurtatuz eta aplikazioa modu malkartsuan itxi dadin saihestuz.

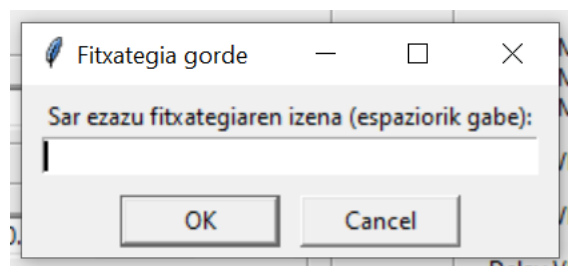
### Save\_n2 funtzioa

Save\_n2 funtzioak egungo konfigurazioak eta optimizazio-emaitzak fitxategi batean gordetzeko aukera ematen dio erabiltzaileari, eta datu horiek etorkizunean atzitzea eta berrrerabiltzea errazten du.

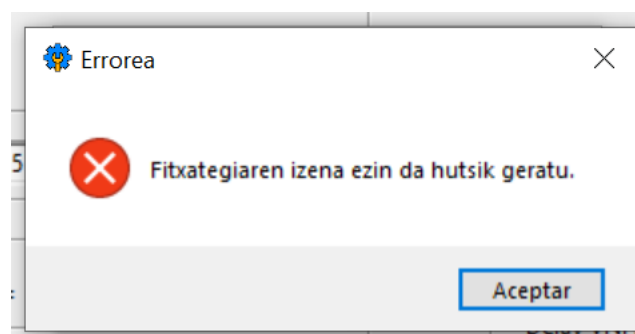
Funtzionamendu orokorra:

- **Artxibo-izena baliozkotzea**: erabiltzaileari eskatzen zaio fitxategi-izen bat sar dezala elkarrizketa-koadro baten bidez. Izena baliozkotu egiten da hutsik ez dagoela eta karaktere alfanumerikoak bakarrik dituela ziurtatzeko.

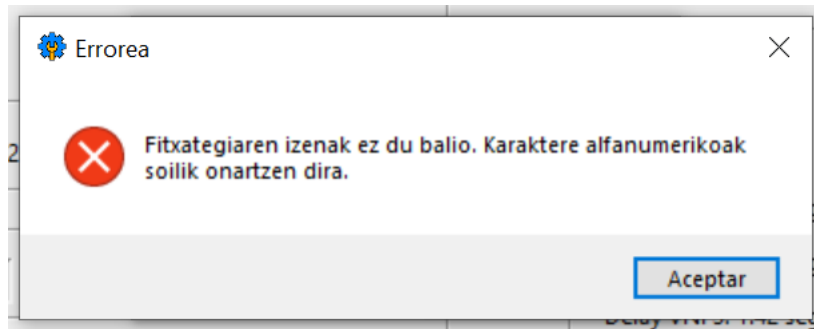
*Irudia 24: izen eskaera.*



*Irudia 25: izena sartu gabe utziz gero ateratzen den errore mezua.*

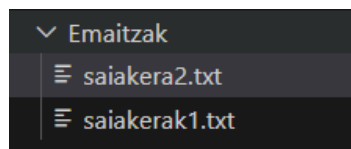


*Irudia 26: izenean karaktere alfanumerikoak soilik ageri ez direnean ateratzen den errore mezua.*



- **Direktorioa sortzea:** direktoriorik ez badago, berariazko direktorio bat sortzen da (Emaitzak), emaitzen fitxategiak gordetzeko.

*Irudia 27: fitxategiak "Emaitzak" izeneko direktorioan gordeta.*



- **Datuak idaztea:** funtzioak hautatutako fitxategian idazten ditu sarrerako aldagaiak eta optimizaziotik lortutako emaitzak, eta informazioa argi zehaztutako ataletan antolatzen du.

*Irudia 28: aurrez gordetako konfigurazioaren eta emaitzen fitxategia.*

```
≡ saiakera2.txt ×
Emitzak > ≡ saiakera2.txt
1 Aldagaiak:
2 mu_str: 2000
3 p11_str: 0.25
4 p12_str: 0.25
5 p21_str: 0.25
6 p22_str: 0.25
7 rhoNodo_str: 0.1
8 rhoSIP_str: 0.1
9 rhoVNF1_str: 0.1
10 rhoVNF2_str: 0.1
11 rhoVNF3_str: 0.1
12 retN1N2_str: 0.006
13 retN1MCX_str: 0.002
14 retN1IMS_str: 0.002
15 retN2N1_str: 0.006
16 retN2MCX_str: 0.002
17 retN2IMS_str: 0.002
18 retMCXN1_str: 0.002
19 retMCXN2_str: 0.002
20 retMCXIMS_str: 0.001
21 retIMSN1_str: 0.002
22 retIMSN2_str: 0.002
23 retIMSMCX_str: 0.001
24
25 Optimizazioaren emaitza:
26 VNF1: MCX
27 VNF2: MCX
28 VNF3: MCX
29 Konputazio denbora: 1.01 segundo
30 Delay VNF1: 1.69 segundo
31 Delay VNF2: 1.42 segundo
32 Delay VNF3: 1.42 segundo
```

Salbuespenen erabilera:

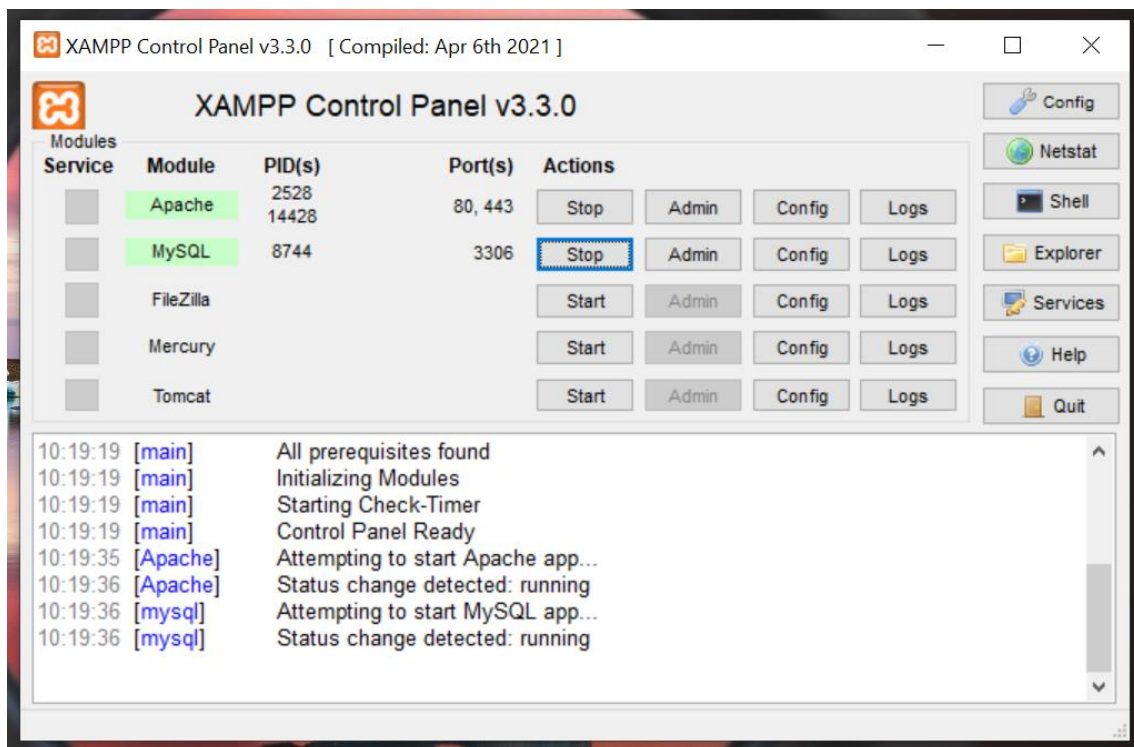
- **IOError edo EOSrro:** fitxategia sortzearekin edo idaztearekin lotutako edozein arazo maneiatzen da, eta errore-mezu bat erakusten da fitxategia ezin bada behar bezala gorde.

- **General Exception Handling:** salbuespen orokorrak atzematea, erabiltzaileari atzeraelikadura emateko, gordetze-eragiketan ustekabeko akatsen bat gertatuz gero.

## 6.2.5. Datu-basearekin konexioa

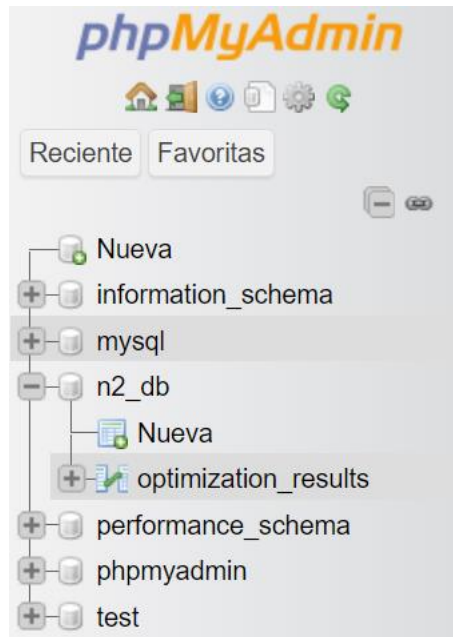
Aplikazioaren frontend-a datu-base batera konektatzen da, MySQL zerbitzariaren bidez kudeatutako MariaDB plataforma erabiliz. MariaDB datu-baseak kudeatzeko sistema gisa erabiltzen da, eta datuak biltegitratzeko beharrezkoak diren taulak sortzeko eta kudeatzeko aukera ematen du. Zerbitzaria martxan jartzeko eta konexioa errazteko, XAMPP Control Panel aplikazioa erabiltzen da, Apache zerbitzaria eta MySQL zerbitzaria abiarazteko. XAMPP Control Panelak zerbitzari lokal baten exekuzioa sinplifikatzen du, web-zerbitzariarentzako Apache eta datu-basearentzako MySQL kudeatuz, eta datu-baserako sarbidea erraztuz phpMyAdmin-en bidez.

*Irudia 29: XAMPP Control Panel.*



Datu-baseak *localhost/phpmyadmin*-etik administratzen dira. phpMyAdmin interfaze grafikoa da, eta datu-baseak modu errazean eta bisualean kudeatzeko aukera ematen du, taulak sortzea, datuak txertatzea eta aldatzea erraztuz.

*Irudia 30: phpMyAdmin datu baseak.*



*Irudia 31: phpMyAdmin optimization\_results taulako aldagaiak.*

id	mu	p11	p12	p21	p22	rhoNodo	rhoSP	rhoVNF1	rhoVNF2	rhoVNF3	retN1N2	retN1MCX	retN1IMS
1	2000	0.25	0.25	0.25	0.25		0.1	0.1	0.1	0.1	0.006	0.002	0.002

retN2N1	retN2MCX	retN2IMS	retMCXN1	retMCXN2	retMCXIMS	retIMS1	retIMS2	retIMSMCX
0.006	0.002	0.002	0.002	0.002	0.001	0.002	0.002	0.001

VNF1	VNF2	VNF3	ComputationTime	delayVNF1	delayVNF2	delayVNF3
3	3	3		1.53	1.69	1.42

Tresna horien konbinazioak ziurtatzen du aplikazioak modu eraginkorrean eta seguruan elkarreragin dezakeela datu-basearekin, optimizazio-datuak biltegitzeko eta berreskuratzeko plataforma sendo bat eskainiz.

Datu-baserako konexioa aplikazioaren eta MariaDBren arteko komunikazioa kudeatzen duten zenbait funtzioen bidez gauzatzen da. Jarraian deskribatuko ditugu interakzio hori erabiltzen duten funtzio nagusiak:

### **connect\_to\_db**

Funtzio honek n2\_db datu-basearekiko konexioa ezartzen du, Python-erako MariaDB konektorea erabiliz. Konexio bat definitzen da erabiltzailearen kredentzialen eta host



lokalaren bidez, erabili nahi den ataka eta datu-basea zehaztuz. Konexioan errore bat gertatzen bada, salbuespen bat atzematen da eta errorearen berri ematen da, programaren exekuzioa gelditu gabe. Funtzio hori funtsezkoa da, konexio segurua eta kontrolatua ematen baitu datu-basearekin, konexio-saiakeran gertatzen den edozein akats modu egokian maneiatzen dela ziurtatuz.

### **data\_to\_db**

Konexioa ezarrita dagoenean, funtzio honek optimizazio-datuak datu-basearen `optimization_results` taulan txertatzen ditu. Taula honek hasierako parametro guztiak eta aplikazioan gauzatutako optimizazioaren emaitzak biltegitratzen ditu. Funtzioak frontend-eko sarreraren balioak ateratzen ditu, float motara bihurtzen ditu eta hiztegi batean biltegitratzen ditu. Funtzioak `optimization_results` taulan txertatzen ditu datuak, SQL injekzioak saihesteko eta datuen osotasuna ziurtatzeko prestatutako kontsulta bat erabiliz. Gainera, datuak txertatzean sor daitekeen edozein salbuespen kudeatzen da, eta, beharrezkoa bada, errore-mezu bat erakusten da.

Garrantzitsua da nabarmentzea `data_to_db` funtzioa optimizazio-prozesua exekutatu ondoren exekutatzen dela, honako kode-lerro honen bidez:

```
command = lambda: [optimization_n2 (), update_result_optimization (), data_to_db ()]
```

Hemen, `optimization_n2` funtzioak optimizazioa egiten du eta emaitzak lortzen ditu. Emaitzak pantailan bistaratzen dira `update_result_optimization` bidez. Azkenik, datuak, optimizazioaren emaitzekin batera, modu seguruan biltegitratzen dira datu-basean `data_to_db` bidez.

## **6.2.6. Direktorioen eta fitxategien egitura**

Proiektuaren direktorioen eta fitxategien egitura aplikazioaren osagai nagusietarako sarbidea errazteko moduan antolatuta dago, scriptak, konfigurazio-datuak, emaitzak eta beste elementu garrantzitsu batzuk argi eta ordenatuta kudeatzeko aukera emanez. Jarraian, proiektuaren direktorioen egitura deskribatzen da:

*/Proiektu finala*

*/Emaitzak*

*Froga1.txt*

*Froga2.txt*

...

*/topologia 2 nodo*

*fullSequence.py*

*vnf01Init.py*

...

*/topologia 3 nodo*

*fullSequence.py*

*vnf01Init.py*

...

*main.py*

*nodo2\_values.txt*

*optimization.ico*

*n2\_topologia.png*

- **Proiektu finala:** proiektuaren erro direktorioa da, aplikazioak funtzionatzeko behar diren fitxategi eta karpeta guztiak dituena.
- **Emaitzak:** karpeta honetan egindako optimizazioen emaitzak dituzten testu-fitxategiak gordetzen dira. Fitxategien izenek egindako proben testuingurua edo izaera islatzen dute, eta errazago identifikatzen eta aztertzen dira.
- **topologia 2 nodo:** direktorio honek bi nodoko topologia simulatzeko eta optimizatzeko Python scripts espezifikoak ditu. *fullSequence.py* fitxategia optimizazioaren fluxu osoa kudeatzen duen script nagusia da; *vnf01Init.py* eta beste fitxategi batzuk, berriz, topologiaren barruan funtzio espezifikoak hasieratzen eta konfiguratzeko dituzte.
- **topologia 3 nodo:** 2 nodo karpetaaren antzekoa, direktorio honetan 3 nodo dituen topologia konplexuago bat simulatzeko erabiltzen diren script-ak daude.

Fitxategien izenek simulazioan eta optimizazioan dituzten rol espezifikoak adierazten dituzte.

- **main.py:** script nagusia da, eta programaren zatien arteko elkarrekintza sustatzen du, interfaze grafikoa, datu-baserako konexioa eta optimizazio-prozesuen exekuzioa barne.
- **nodo2\_values.txt:** hasiera-aldagaiak eta optimizazio-emaizak aldi baterako biltegitratzeko erabiltzen den testu-fitxategi bat, saioen arteko datu-transferentzia eta ondorengo analisietarako beharrezkoa den informazioaren iraunkortasuna erraztuz.
- **optimization.ico:** aplikazioan erabiltzailearen interfaze grafikoaren estetika hobetzeko erabiltzen den ikono pertsonalizatua.
- **n2\_topologia.png:** aplikazioan erabiltzen den nodo-sarearen topologia irudikatzen duen irudia, optimizazio-emaizak konfiguratzeko eta bistaratzean erreferentzia bisual erabilgarria eskainiz.

Proiektuaren egitura antolatua bermatzen du osagai bakoitza leku egokian egotea, eta horrek erraztu egiten du nabigazioa eta kodearen eta datuen kudeaketa. Antolaketa horrek, halaber, aldatu edo aztertu nahi duten aplikazioaren zati espezifikoak azkar identifikatzen laguntzen die sistemaren garatzaileei eta erabiltzaileei. Gainera, banakako direktorioetako topologiaren eta emaitzen bereizketa logikoak lan-fluxuaren argitasuna hobetzen du, sistemaren integrazioa eta arazketa maneigarriagoa eta eraginkorragoa eginez.

### 6.2.7. Lortutako emaitzak

Aplikazio honetan proposatutako ereduaren erabilgarritasuna eta eraginkortasuna frogatzeko, atal honetan ereduaren zenbakiz ebatzi eta lortutako emaitzak interpretatzen dituzten adibide batzuk aurkezten dira. Adibide horiek aukera ematen dute ikusteko nola eragiten duen erabiltzaileen kokapenak eta nodoetan erabilgarri dagoen gaitasun konputazionalak VNF funtzioen banaketan eta prozesamenduan.

Azterketa bi agertoki nagusitan egiten da, bi nodoko topologia erabiliz, eta, ondoren, hiru nodoko topologia batera hedatzen da, sistemaren errendimenduan duen eragina alderatzeko.

Aztertutako kasu guztietan, ereduak sarrerako datu hauek erabiltzen ditu:

- **VNF zerbitzatzeko batez besteko denbora:** parametro honek definitzen du sare-funtzio birtualizatu batek datu-fluxu bat prozesatzeko batez beste behar duen denbora.
- **Nodo pare bakoitzaren arteko sare-latentzia:** datu-pakete batek nodo batetik bestera bidaiatzeko behar duen denbora da latentzia. Funtsezkoa da sareko informazio-fluxuaren eraginkortasuna zehazteko.
- **N1 eta N2 nodoetara konektatutako erabiltzaileen artean fluxuak transmititzeko probabilitateak:** balio horiek zehazten dute nola banatzen diren datu-fluxuak sarearen bidez, eta zuzenean eragiten diote nodo bakoitzaren prozesamendu-kargari.
- **Erabiltzaile bakoitzak sortutako trafikoa:** hau da erabiltzaile bakoitzak sarera bidaltzen duen datu-bolumena, eta horrek eragina du nodo bakoitzaren guztizko kargan eta, ondorioz, VNFen prozesamenduaren eraginkortasunean.
- **VNF bakoitzaren karga-ratioa edo rho balioa:** parametro hori  $\rho_i = \frac{\lambda_i}{\mu_i}$  gisa definitzen da, non VNF<sub>i</sub>-ra iristen diren fluxuen tasa ( $\lambda_i$ ) eta prozesatzeko gaitasuna ( $\mu_i$ ) diren. Ratio hori funtsezkoa da VNF bat edozein unetan fluxu bat prozesatuz okupatuta egoteko probabilitatea zehazteko.

Garrantzitsua da adieraztea, 3GPPk LTEri buruzko MCPTTrako hainbat errendimendu-adierazle (KPI) definitu dituen arren, hemen aurkeztutako adibideetan KPI adierazle horien balio ez-errealak erabiltzen direla. Hala ere, horrek ez du eraginik ereduaren ebaluazioaren ondorioetan, helburua sistemaren dinamika hainbat baldintzatan irudikatzea baita, eta ez zenbakizko emaitzen zehaztasuna.

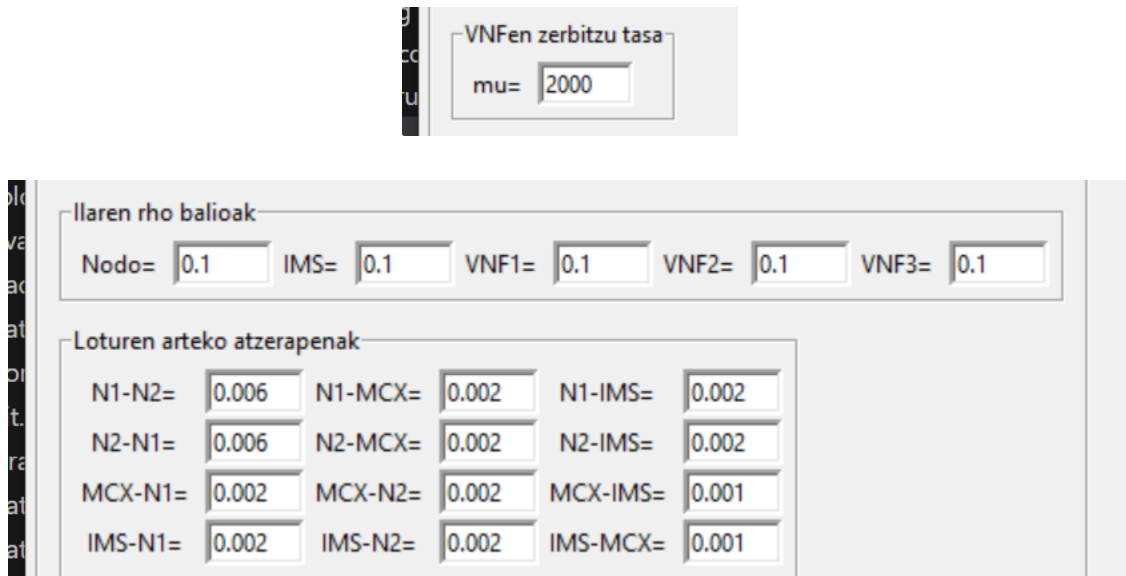
## 1. adibidea: 2 nodoen topologia

Adibide honetan, erabiltzaileak bi sARBIDE-nodoren artean banatzeak eta fluxuak transmititzeko probabilitateek VNF funtzioen esleipen optimoari nola eragiten dioten

aztertuko dugu. Bi kasu desberdin aztertuko ditugu, VNFak sare-baldintza desberdinetan hedatzeko estrategia nola aldatzen den azaltzeko.

Kasu bietan aldagai hauek mantenduko dira:

***Irudia 32: mantentzen diren aldagaiak.***



VNFen zerbitzu tasa  
mu= 2000

Ilaren rho balioak  
 Nodo= 0.1 IMS= 0.1 VNF1= 0.1 VNF2= 0.1 VNF3= 0.1

Loturen arteko atzerapenak

N1-N2=	0.006	N1-MCX=	0.002	N1-IMS=	0.002
N2-N1=	0.006	N2-MCX=	0.002	N2-IMS=	0.002
MCX-N1=	0.002	MCX-N2=	0.002	MCX-IMS=	0.001
IMS-N1=	0.002	IMS-N2=	0.002	IMS-MCX=	0.001

**1. kasua: erabiltzaileen bidezko banaketa**

Lehenengo kasuan, erabiltzaileen banaketa ekitatiboa da bi sarbide-nodoen artean, non erabiltzaileen %50 N1 nodoari konektatuta dauden eta beste %50 N2 nodoari. Nodo horietara konektatutako erabiltzaileen artean fluxuak transmititzeko probabilitateak honako hauek dira:

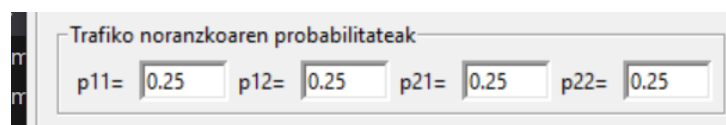
$p_{11} = 0.25$ : N1etik N1era transmititzeko probabilitatea.

$p_{12} = 0.25$ : N1etik N2ra transmititzeko probabilitatea.

$p_{21} = 0.25$ : N2tik N1era transmititzeko probabilitatea.

$p_{22} = 0.25$ : N2tik N2ra transmititzeko probabilitatea.

***Irudia 33: fluxuak transmititzeko probabilitateak***



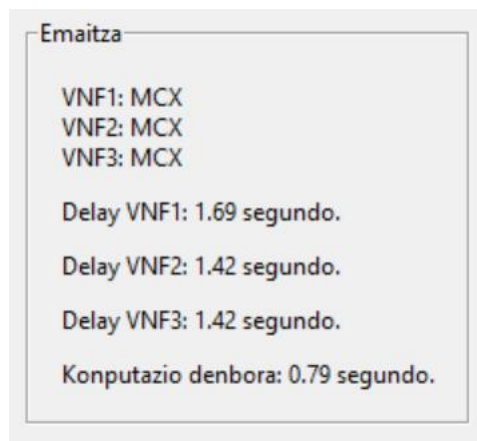
Trafiko noranzkoaren probabilitateak

p11= 0.25 p12= 0.25 p21= 0.25 p22= 0.25

VNFen hedapena iterazioetan:

1. iterazioa: VNF1 funtzioa MCX zerbitzariari esleitzen zaio.
2. iterazioa: VNF2 funtzioa MCX zerbitzariari esleitzen zaio.
3. iterazioa: VNF3 funtzioa MCX zerbitzariari esleitzen zaio.

***Irudia 34: 1. kasuaren emaitzak.***



Emaitzak interpretatuz gero kasu honetan, ikusten dugu birtualizatutako sare-funtzioen hedapen optimoa zerbitzu guztiak MCX zerbitzarian zentralizatzean datzala. Estrategia horren arrazoa erabiltzaileen banaketa ekitatiboa da, eta horrek sarearen latentzia minimizatzen du eta baliabideen erabilera optimizatzen du, VNF guztiak nodo zentral bakar batean mantentzen baititu. Zentralizazio hori eraginkorra da, nodoen arteko datu-transferentziaren beharra murrizten baitu eta MCX zerbitzarian erabilgarri dagoen gaitasun konputazionala optimizatzen baitu.

Sarearen portaera uniformeak, fluxuak transmititzeko probabilitate berdinek eraginda, justifikatu egiten du funtzioak puntu bakar batean kontzentratzea, eta errendimendua maximizatzen du datuak prozesatzean botila-leparik sortu gabe.

**2. kasua: erabiltzaileen banaketa desberdina**

Bigarren kasuan, erabiltzaileen banaketa aldatzen da, non erabiltzaileen %70 N1 nodoari konektatuta dauden eta %30 N2 nodoari. Fluxuak transmititzeko probabilitateak nabarmen aldatzen dira, eta trafiko-karga handiagoa islatzen dute N1ean:

$$p_{11} = 0.7: N1etik N1era transmititzeko probabilitatea.$$

$p_{12} = 0$ : N1etik N2ra transmititzeko probabilitatea.

$p_{21} = 0.2$ : N2tik N1era transmititzeko probabilitatea.

$p_{22} = 0.1$ : N2tik N2ra transmititzeko probabilitatea.

***Irudia 35: fluxuak transmititzeko probabilitateak.***

Trafiko noranzkoaren probabilitateak  
p11= 0.7 p12= 0 p21= 0.2 p22= 0.1

VNFen hedapena iterazioetan:

1. iterazioa: VNF1 funtzioa MCX zerbitzariari esleitzen zaio.
2. iterazioa: VNF2 funtzioa MCX zerbitzariari esleitzen zaio.
3. iterazioa: VNF3 funtzioa N1 nodoari esleitzen zaio.

***Irudia 36: 2. kasuko emaitzak.***

Emitza  
VNF1: MCX  
VNF2: MCX  
VNF3: N1  
Delay VNF1: 1.68 segundo.  
Delay VNF2: 1.42 segundo.  
Delay VNF3: 1.42 segundo.  
Konputazio denbora: 0.88 segundo.

Emaitzak interpretatuz gero egoera horrek erakusten du erabiltzaileen banaketa desorekatuak VNFak hedatzeko estrategia aldatzen duela. Zerbitzu guztiak MCX zerbitzarian zentralizatu beharrean, ereduak honela banatzen ditu funtzioak:

VNF1 eta VNF2: MCX zerbitzarian zentralizatuta egoten dira, eta horrek prozesamendu eraginkorra eskaintzen du trafiko gehienarentzat, nodoen arteko edo zerbitzari nagusiaren barruko transmisioak barne hartzen baititu.

VNF3: N1 nodoan hedatzen da, trafiko lokalaren karga handia eta barne-transmisioaren probabilitate handia direla eta ( $p_{11}$ ). Horrek latentzia murrizten du eta erantzun-denbora hobetzen du N1era konektatutako erabiltzaileentzat.

2. kasuan, ereduaren malgutasuna eta moldagarritasuna nabarmentzen dira, trafiko-karga uniformeki banatuta ez dagoen sareko konfigurazioetara egokitzeko. VNF3a eskari handieneko nodorantz deszentralizatzeko aukera ematen duenez, baliabideen balantze eraginkorragoa eta sistemaren errendimendua hobetzea lortzen da.

Horrela bada, 1. adibideak erakusten du erabiltzaileen kokapenak eta sarearen egiturak eragin zuzena dutela sareko funtzio birtualizatuen hedapen optimoan. 1. kasuan, MCX zerbitzari bateko zentralizazioa da irtenbiderik eraginkorrena; 2. kasuan, berriz, VNFen banaketa estrategikoagoa behar da, sistemaren errendimendua optimizatzeko.

Azterketa honek hedapen-estrategiak sarearen baldintza espezifikoetara egokitzearen garrantzia azpimarratzen du, eta hori kritikoa da baliabide konputazionalen erabilera maximizatzeko eta erabiltzailearen esperientzia hobetzeko. Gainera, nabarmentzekoa da topologia eta trafikoa kontuan hartzearen garrantzia VNFen hedapenari buruzko erabakiak hartzerakoan, sarearen eragiketa optimoa eta eraginkorra lortzeko.

## **2. adibidea: Hiru Nodoen Topologiarekin alderatzea**

Bigarren adibide honetan, hirugarren nodo bat duen sare-topologia batek VNF funtzioen hedapenean eta sistemaren errendimendu orokorrean nola eragiten duen ebaluatuko da. Topologiari konplexutasuna gehitzean, VNFen esleipenaren desberdintasunak eta soluzio optimo bat lortzeko behar den denbora konputazionalaren efektuak aztertuko ditugu. Sare handiagoak optimizatzean sortzen diren zailtasunak eta erronka horiek gainditzeko irtenbide posibleak ere eztabaidatzen ditugu.

### **1. kasua: erabiltzaileen bidezko banaketa**

Lehenengo kasu honetan, suposatzen da erabiltzaileak zuzen banatuta daudela hiru nodoen artean (N1, N2 eta N3), eta horrek esan nahi du nodo bakoitzak gutxi gorabehera erabiltzaileen karga-kopuru bera jasaten duela. Zerbitzuaren batez besteko denbora 0,5ms-tan mantentzen da, nodoen rho balioak bezala, hauek 0,1ean mantentzen baitira.



N1, N2 eta N3 nodoetara konektatutako erabiltzaileen artean fluxuak transmititzeko probabilitateak modu desberdinean esleitu dira, eta guztira 9 probabilitate daude, non horietako bat besteak baino pixka bat handiagoa den:

$$p_{11} = 0,2$$
$$p_{12} = p_{13} = p_{21} = p_{22} = p_{23} = p_{31} = p_{32} = p_{33} = 0,1$$

Gainera, sare –  
latentzia berriak sartzen dira nodoen artean, baita nodoen eta MCX zerbitzariaren artean ere:

$$\delta(N1, N3) = \delta(N2, N3) = \delta(N3, N1) = \delta(N3, N2) = 6 \text{ ms}$$
$$\delta(MCX, N3) = \delta(IMS, N3) = \delta(N3, MCX) = \delta(N3, IMS) = 2 \text{ ms}$$

Baldintza hauekin, optimizazioaren emaitzek adierazten dute VNF funtzio guztiak (VNF1, VNF2 eta VNF3) MCX zerbitzariari esleitzen zaizkiola.

Trafiko-mota bakoitzerako batez besteko atzerapenak hauek dira:

$$\text{Delay VNF1} = 1,81 \text{ ms}$$

$$\text{Delay VNF2} = 1,36 \text{ ms}$$

$$\text{Delay VNF3} = 1,36 \text{ ms}$$

### *Irudia 37: 1. kasuko emaitzak.*

```
VNF1: MCX
VNF2: MCX
VNF3: MCX
Konputazio denbora: 1225.21
Delay VNF1: [1.80698503]
Delay VNF2: [1.36255015]
Delay VNF3: [1.36130656]
```

## **2. kasua: erabiltzaileen banaketa ez-ekitatiboa**

Bigarren egoera horretan, erabiltzaileen banaketa ez da zuzena: erabiltzaileen %70 N1 nodora konektatuta daude, eta gainerako %30a N2 eta N3 artean banatzen da. Fluxuak transmititzeko probabilitateak honela definitzen dira:

$$p_{11} = 0,7$$

$$p_{12} = p_{13} = 0$$

$$p_{21} = p_{22} = p_{23} = p_{31} = p_{32} = p_{33} = 0,05$$

Gainerako aldagaiak aurreko kasukoak berdinak dira. Baldintza horietan, emaitzek VNF esleipen hauek adierazten dituzte:

VNF1 MCX zerbitzariari esleitzen zaio.

VNF2 N1 nodoari esleitzen zaio.

VNF3 MCX zerbitzariari esleitzen zaio.

Trafiko-mota bakoitzerako batez besteko atzerapenak hauek dira:

Delay VNF1 = 1,74 ms

Delay VNF2 = 1,39 ms

Delay VNF3 = 1,39 ms

***Irudia 38: 2. kasuko emaitzak.***

```
VNF1: MCX
VNF2: N1
VNF3: MCX
Konputazio denbora: 727.76
Delay VNF1: [1.74043529]
Delay VNF2: [1.39583333]
Delay VNF3: [1.39583333]
```

Bi kasuetan optimizatzeko konputazio-denborak alderatzean, 3 nodoen topologiako solverren konbergentzia alde nabarmena ikusten da. Bi nodoen topologiarekin alderatuta, non konputazio-denborak segundo batekoak diren gutxi gorabehera, 3 nodoko topologiaren konplexutasun gehigarriak denborak nabarmen handitzea eragiten du:

- 1. kasua: 1225,21 segundo (20 minutu, gutxi gorabehera)
- 2. kasua: 727,76 segundo (12 minutu, gutxi gorabehera)

Prozesatze-denbora horiek oso altuak dira, eta onartezintzat jo daitezke horrelako aplikazio baterako. Konputazio-denboraren igoera hori, hein handi batean, nodo kopuruaren eta lotutako konplexutasunen hazkundearen ondorio da.

Gutxienez konbergentzia adierazten duen soluzio bat lortzeko, doikuntzak egin ziren solVERRAREN murrizketetan. Minimize funtzioan *trust-constr* metodoa erabiliz, *xtol* eta *maxiter* parametroak aldatu ziren:

- **xtol (aldagai independenteetako tolerantzia):** parametro honek sarrerako aldagaien konbergentziarako tolerantzia definitzen du. Kasu honetan, 1e-5eko balioa ezartzen da, eta horrek konbergentzia laxoagoa ahalbidetzen du lehenetsitako 1e-8ko balioarekin alderatuta.
- **maxiter (iterazio kopuru maximoa):** parametro honek algoritmoa mugarik gabe exekutatzea aurreikusten du. 10000era doitu da, eta lehenetsitako balioa (1000) baino iterazio-kopuru handiagoa ahalbidetu du.

*Irudia 39: options aukera, minimize funtzioaren barruan.*

```
options={'xtol': 1e-5, 'maxiter': 10000})
```

Doikuntza horien helburua da algoritmoak soluzio optimo batekin bat egin ahal izatea, nahiz eta konputazio-denbora luzeagoak izan.

Azterketa honek erakusten du, sarearen topologia konplexuagoa bihurtzen den heinean, konputazio-denbora nabarmen handitzen dela. Horrek adierazten du bi nodo baino gehiagoko topologietarako optimizazio-alternatibak esploratu behar direla, ereduaren zehaztasuna arriskuan jarri gabe konputazio-eraginkortasuna hobetu dezaketen metodoak bilatuz.

## 7. Lan-plana

---

Dokumentuaren atal honetan, proiektua gauzatzeko jarraitutako lan-plana aurkezten da. Plangintza xehatuak funtsezko hiru azpiatal ditu: lantaldea eta baliabide materialak, lan-paketeak eta zereginak, eta jardueren denbora epea bistaratzen duen Gantt diagrama.

### 7.1. Lantaldea eta baliabide materialak

Proiektu hau garatu duen lantaldea honako kide hauek osatzen dute:

*Table 1: proiektuko kideen zerrenda.*

Kodea	Izena	Erantzukizuna	Rola
I1	Luis Zabala Alberdi	Senior ingeniaria	Proiektuaren gainbegiraketa eta zuzenketa
I2	Aratz Longo Bilbao	Junior ingeniaria	Proiektua gauzatzea

Proiektua gauzatzeko behar diren baliabide materialei dagokienez, taula honetan zehazten dira:

*Table 2: proiektuko baliabide materialen zerrenda.*

Kodea	Materiala	Kopurua
PC	Ordenagailua	1

### 7.2. Lan-paketeak eta zereginak

Proiektu baten plangintza hainbat lan-paketek (LP) osatzen dute, eta pakete horiek, aldi berean, hainbat zereginetan banakatzen dira (A). Ondoren, lan-pakete eta ataza bakoitza deskribatuko da, haien xehetasunekin eta epeekin batera.

#### ***LP1 - Proiektua kudeatzea eta dokumentazioa garatzea***

Fase honek proiektu osoan zehar egingo diren etengabeko zereginak biltzen ditu, garapen egokia eta haren dokumentazioa ziurtatuz. Ingeniari juniorren eta seniorren arteko

elkarreraginak aurrerapena monitorizatua izatea eta edozein arazori azkar heltzea bermatzen du.

**A11 - Proiektuaren garapenaren jarraipena:** jarraipenaren bidez, garapenaren aurrerapenaren eta aurkitutako arazoen berri emango da. Gai horiek lantaldearekin egingo diren bileretan aztertuko dira.

Iraupena: 25 ordu proiektu osoan zehar.

**A12 - Dokumentazioa garatzea:** proiektuaren dokumentazioa osatuko da, eta garapena eta ondorioak deskribatuko dira. Proiektua amaitzean, jasotako informazio guztiarekin amaierako txosten bat egingo da. Zeregin hori proiektua garatzearekin batera egiten da.

Iraupena: 40 egun.

Hauek dira fase honetarako behar diren baliabideak:

- I1: 20 ordu.
- I2: 100 ordu.

### ***LP2 - Proiektuaren definizioa***

Fase hau proiektuaren hasieran gauzatzen da, geroko lan osorako oinarriak eta jarraibideak ezarriz. Proiektuaren helburuak, irismena eta funtsezko kontzeptuak zehazten dira, eta hori erabakigarria da hurrengo etapetarako.

**A21 - Helburuak zehaztea:** proiektuaren helburuak ezartzen ditu, eta azken emaitza ona izango dela bermatzeko bete behar diren ezaugarriak eta irizpideak definitzen ditu.

Iraupena: 15 egun.

**A22 - Alternatibak aztertzea:** proiekturako garapen-software egokiena ebaluatzea eta hautatzea, merkatuan eskuragarri dauden aukeren analisi konparatibo baten bidez.

Iraupena: 15 ordu.

Hauek dira fase hau osatzeko behar diren baliabideak:

- I1: 15 ordu.
- I2: 50 ordu.

### ***LP3 - Aplikazioaren diseinua***

Fase honetan, programaren egitura-diseinua garatzen da, bai backend mailan, bai frontend mailan, beharrezko datu-baseen plataforma txertatzeaz gain. Horren barruan sartzen dira optimizazio-funtzioak, erabiltzaile-interfazerako liburutegiak eta datu-base eraginkor bat hautatzea eta erabiltzea.

**A31 - Optimizaziorako funtzioak:** solver egokia hautatzea eta horren inguruan backend-a garatzea, eta behar bezala funtzionatzeko beharrezkoak diren aldagaiak eta murrizketak definitzea.

Iraupena: 40 egun.

**A32 - Liburutegiak eta hedapenak:** programaren beharretara egokitutako erabiltzaile-interfazea garatzea ahalbidetuko duen liburutegia hautatzea. UIa optimizazio-programarako zuzeneko konexioarekin garatzen da.

Iraupena: 30 egun.

**A33 - Datu-baseen plataforma txertatzea:** datu-baseen plataforma hautatzea eta inplementatzea, eta informazio optimizatua biltegiratzeko eta kudeatzeko beharrezko funtzioak sortzea, programa nagusia gehiegi kargatu gabe.

Iraupena: 5 egun.

Hauek dira fase hau osatzeko behar diren baliabideak:

- I1: 25 ordu.
- I2: 150 ordu.

## **7.3. Gantt diagrama**

Dokumentuaren atal honetan, proiektuaren denbora-plangintza Gantt diagrama baten bidez deskribatzen da. Proiektuaren hasiera-data 2024ko maiatzaren 1a da, eta amaiera-data, berriz, 2024ko uztailaren 26a.



## 8. Aurrekontua

---

Dokumentuaren atal honetan, proiektu honen garapenak eragindako kostuen laburpena egingo da. Kostuak kalkulatzeko, giza baliabideen eta baliabide materialen kostuak hartuko dira kontuan. Kostu horiek kalkulatzeko, "Lan-plana" atalean zehaztutako ordurak hartuko da kontuan.

### 8.1 Giza baliabideak

"Lan Plana" atalean azaldu den bezala, proiektu hau bi langilek eraman dute aurrera: ingeniari senior batek eta ingeniari junior batek. Hori kontuan hartuko da aurrekontua egiterakoan, ingeniari seniorren orduko tasa handiagoa baita. Ondorioz, hurrengo taulan giza baliabideen kostuen banaketa aurkezten da, eta bertan langile bakoitzaren orduko tasa, plangintzan itundutako ordurak eta azken kostu osoa zehazten dira.

*Table 3: giza-baliabideen kostua.*

Kodea	Erantzukizuna	Orduko tasa	Ordu kopurua	Guztira
I1	Senior ingeniaria	60 €/h	30 h	1.800 €
I2	Junior ingeniaria	25 €/h	300 h	7.500 €
<b>Guztira</b>				<b>9.300 €</b>

### 8.2. Baliabide materialak

"Lan-plana" atalean zehazten diren baliabide materialak beharrezkoak izan dira proiektua gauzatzeko. Baliabide materialen kostua kalkulatu ahal izateko, amortizazioa aztertu da.

Amortizazioa kalkulatzeko, ekipoaren hasierako kostua eta bizi-zikloa hartzen dira kontuan. Jarraian, taula bat ageri da, proiektuan erabilitako baliabide materialaren amortizazioaren kalkulua zehazten duena:



**Table 4: baliabide materialen kostua.**

Kodea	Izena	Kostua	Bizitza erabilgarria	Urteroko amortizazioa	Proiektuaren iraupena	Proiektuaren amortizazioa
PC	Ordenagailua	1.100 €	5 urte	220 €	3 hilabete	55 €
<b>Guztira</b>						<b>55 €</b>

Gastu horiez gain, ez dago beste gasturik, hala nola azpikontratazioak edo software-lizentziak. Beraz, garapenaren guztizko gastua kalkulatzeko, atal honetan aintzat hartutako gastu guztiak hartuko dira kontuan: lanorduak eta baliabide materialen amortizazioak.

**Table 5: kostuak guztira.**

Kontzeptua	Guztira
Barne-orduak	9.300 €
Amortizazioak	55 €
<b>Guztira</b>	<b>9.355 €</b>

Beraz, barneko orduak eta amortizazio-kostuak kontuan hartuta, proiektuak guztira 9.355€-ko gastua duela kalkulatzen da.

## 9. Ondorioak

---

Proiektu honek 5G sareetan VNF funtzioen kokapena optimizatzeko gai den software bat garatzea lortu du eta Python eta Tkinter erabiliz, erabiltzaile-interfaze funtzional bat inplementatu da. Garapenean, azpiko eredu matematikoa ulertzeko, teknologia berrietako gaitasunak eskuratzeko eta Misio Kritikoko sareen azpiegitura aztertzeke helburuak bete dira.

Hala ere, proiektuaren helburu garrantzitsu batzuk ez dira lortu. Aurrerapenak aurrerapen, ezin izan zen 3 nodoen topologia arrakastaz ezarri, optimizazio-prozesuan konbergentzia-arazoak izan zirelako. Arazo horien ondorioz, sistemak ezin izan zuen eraginkortasunez funtzionatu topologia konplexuago batean, eta horrek mugatu egiten du egungo aplikazioa. Gainera, inplementazioa komunikazio indibidualean bakarrik zentratu zen, taldeko komunikazioaren funtzionaltasunari heldu gabe, funtsezkoa baita MCPTT sistema benetan erabiltzeko. Sistema horretan, mezu batek hainbat hartzailerengana iristeko aukera izan behar du aldi berean. Horrek esan nahi du egungo soluzioak ez duela erabat adierazten aplikazioaren potentziala erabilera errealeko egoeretan, non talde-komunikazioa kritikoa den.

Etorkizunera begira, interes handiko hainbat arlo daude proiektu hau hedatzeko. Optimizazio-prozesuetarako adimen artifiziala ezartzeak sistemaren eraginkortasuna alda lezake, eta, hala, dinamikoki egokitu daiteke sare-baldintza aldakor eta konplexuetara. Ikuspegi horrek konbergentzia-arazoak gainditzen eta taldekako komunikazioaren eraginkortasuna nabarmen hobetzen lagun lezake, 5G sareetako aplikazio aurreratuagoi eta sendoagoi atea irekiz. Gainera, optimizazio-metodo alternatiboak eta algoritmo moldagarriak arakatzeak konponbide sendoagoak eskain ditzake sistemaren egungo mugetarako.

## 10. Bibliografía

---

- [1] F.J. Ibañez, «Estudio de las tecnologías SDN y NFV,» Trabajo Fin de Grado, Universitat Oberta Catalunya (UOC), 2016.
- [2] ETSI, «Network Functions Virtualisation (NFV); Management and Orchestration,» 2014.
- [3] L. Zabala, «Modelos matemáticos basados en consumos computacionales para el estudio de rendimiento de sondas de análisis de tráfico en redes de datos,» Tesis doctoral, Euskal Herriko Unibertsitatea (UPV/EHU), 2023.
- [4] 3GPP, «MCPTT 3GPP Standard Specifications,» 2014.
- [5] GUROBI OPTIMIZATION, «Gurobi Optimizer Reference Manual,» <https://www.gurobi.com/documentation/current/refman/index.html>, accessed: 2024-07-26.
- [6] IBM, «IBM ILOG CPLEX Optimization Studio,» <https://www.ibm.com/es-es/products/ilog-cplex-optimization-studio>, accessed: 2024-07-26.
- [7] FICO, «FICO Xpress Solver,» <https://www.fico.com/es/products/fico-xpress-solver>, accessed: 2024-07-26.
- [8] GNU, «GNU Linear Programming Kit,» 2010, <https://kam.mff.cuni.cz/~elias/optimed/glpk.pdf>, accessed: 2024-07-26.
- [9] COIN, «COIN-OR,» <https://www.coin-or.org/>, accessed: 2024-07-26.
- [10] Scipy, «Scipy,» <https://scipy.org/>, accessed: 2024-07-26.
- [11] Scipy, «Minimize,» [https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html), accessed: 2024-07-26.

- [12] R.S. Sutton, A.G. Barto, «Reinforcement Learning: An Introduction,» MIT press, 2018.
- [13] Python, «Python,» <https://www.python.org/>, accessed: 2024-07-26.
- [14] Python, «tkinter — Python interface to Tcl/Tk,» <https://docs.python.org/es/3/library/tkinter.html>, accessed: 2024-07-26.
- [15] Scipy, «`minimize(method='trust-constr')`,» <https://docs.scipy.org/doc/scipy/reference/optimize.minimize-trustconstr.html#optimize-minimize-trustconstr>, accessed: 2024-07-26.
- [16] Scipy, «`minimize(method='L-BFGS-B')`,» <https://docs.scipy.org/doc/scipy/reference/optimize.minimize-lbfgsb.html#optimize-minimize-lbfgsb>, accessed: 2024-07-26.
- [17] Scipy, «`minimize(method='Nelder-Mead')`,» <https://docs.scipy.org/doc/scipy/reference/optimize.minimize-neldermead.html#optimize-minimize-neldermead>, accessed: 2024-07-26.
- [18] Scipy, «`minimize(method='TNC')`,» <https://docs.scipy.org/doc/scipy/reference/optimize.minimize-tnc.html#optimize-minimize-tnc>, accessed: 2024-07-26.
- [19] S. Agarwal, F. Malandrino, C.F. Chiasserini, et al. «VNF Placement and Resource Allocation for the Support of Vertical Services in 5G Networks,» IEEE/ACM Transactions on Networking, vol. 27, 1, pp. 433-446, 2019.