



Universidad del País Vasco    Euskal Herriko Unibertsitatea



## Informatika Ingeniaritzako Gradua Konputazioa

Gradu Amaierako Proiektua

---

# **Caching kooperatiboa Sailkatzaile Bayesiarrak eta algoritmo jaleak erabiliz**

---

Egilea  
*Beñat Azpiazu*



2016



---

## Laburpena

---

Gaur egungo azpiegitura kritikoak, adibidez energia, gasa, ura, garraioa edo osasuna, interneteko konektagarritasunaren oso menpeko bihurtzen ari dira. Beraz, konektagarritasun horrek eragin handia du ekonomian eta gizartean. Garrantzitsua izango da teknologia sektore guztietara eramatea eta baliabideen erabilpen eraginkor bat egitea.

Ahalegin horretan, CogNet europar proiektuaren barnean, cache kooperatiboen teknologiarik adimena ematea proposatu dugu, Machine learning-eko teknikak erabiliz, erabaki azkar eta adimentsuak eman ditzan. Oraingoz probak egiteko azpiegiturarik ez dagoela aprobetxatuz, simulatzaile bat sortu da gure algoritmoen eragina frogatzeko.

Idea hau gauzatzeko, edukiak ospe mailaren arabera sailkatuko dituen sailkatzaile bayesiar bat erabiliko dugu. Azkenik, algoritmo jale bat aplikatuko dugu objektuak non gordeko ditugun zehazteko.



---

# Gaien aurkibidea

---

<b>Laburpena</b>	<b>i</b>
<b>Gaien aurkibidea</b>	<b>iii</b>
<b>Irudien aurkibidea</b>	<b>vii</b>
<b>Taulen aurkibidea</b>	<b>ix</b>
<b>1 Sarrera</b>	<b>1</b>
1.1 Testuingurua . . . . .	2
1.1.1 Vicomtech-IK4 zentroa . . . . .	2
1.1.2 CogNet Europar Proiektua . . . . .	4
<b>2 Kudeaketa</b>	<b>9</b>
2.1 Helburuak eta motibazioa . . . . .	9
2.2 Plangintza . . . . .	10
2.2.1 Atazen identifikazioa . . . . .	10
2.2.2 Lanaren Deskonposaketa Egitura (LDE) . . . . .	11
2.2.3 Planifikatutako egutegia . . . . .	11
2.3 Lan Metodologia . . . . .	12
2.3.1 Kudeaketa eta Antolaketa . . . . .	12

iii

---

2.3.2	Emangarriak . . . . .	12
2.3.3	Artxiboen kudeaketa . . . . .	14
2.4	Arriskuen Kudeaketa . . . . .	14
2.5	Baliabideaen Kudeaketa . . . . .	15
2.5.1	Giza Baliabideak . . . . .	15
2.5.2	Baliabide materialak . . . . .	15
<b>3</b>	<b>Ikerketa</b>	<b>17</b>
3.1	Artearen Egoera . . . . .	17
3.1.1	5G Sareak . . . . .	17
3.1.2	Hodeiaren horniketaren Sailkapena . . . . .	21
3.1.3	Cache Algoritmoak . . . . .	22
3.2	Garapen tresnen analisia . . . . .	25
<b>4</b>	<b>Proiektuaren garapena</b>	<b>29</b>
4.1	Arazoaren deskribapena . . . . .	29
4.1.1	Sarrera . . . . .	29
4.1.2	Arkitektura . . . . .	30
4.1.3	Arazoaren formulazio matematikoa . . . . .	32
4.2	Metodoa . . . . .	33
4.2.1	Hautapen prozesua: Edukiak ospe mailaren arabera sailkatu . . . . .	35
4.2.2	Helburu funtzioa: Kokalekua Bilatu . . . . .	37
4.2.3	Egitura . . . . .	37
4.3	Proba Ingurunea: Simulatzaila . . . . .	37
4.3.1	Harien arteko komunikazioa . . . . .	42
4.3.2	Egitura . . . . .	43
4.4	Emaitzak . . . . .	46

---

4.4.1	Ospe Banaketaren arabera . . . . .	47
4.4.2	Cache Tamainen arabera . . . . .	49
4.4.3	Denbora Leihoaren arabera . . . . .	51
<b>5</b>	<b>Difusioa</b>	<b>53</b>
5.1	Extended Abstract . . . . .	53
5.2	EuCNC-n Hitzaldia . . . . .	54
<b>6</b>	<b>Ondorioak</b>	<b>57</b>
6.1	Proiektuaren ondorioak . . . . .	57
6.2	Etorkizunerako lana . . . . .	58
6.2.1	Sailkatzailea . . . . .	58
6.2.2	Kokapena . . . . .	58
6.2.3	Proba Ingurunea . . . . .	59
6.3	Ondorio pertsonalak . . . . .	59
<b>Eranskinak</b>		
<b>A</b>	<b>Extended Abstract-a</b>	<b>63</b>
<b>Bibliografia</b>		<b>67</b>





---

## Irudien aurkibidea

---

1.1	<i>Vicomtech-IK4 zentroaren logoa.</i>	3
1.2	<i>CogNet Proiektuaren balio proposamena.</i>	5
2.1	<i>Proiektuko Lanaren Deskonposaketa Egitura</i>	12
2.2	<i>Proiektuaren Gantt diagrama</i>	13
3.1	<i>5GPPP-aren etorkizuneko estimazioak.</i>	18
3.2	<i>SDN sareen geruzen errepresentazioa.</i>	20
3.3	<i>Hodeiaren horniketa dinamikoaren mapa.</i>	22
3.4	<i>FIFO eta LIFO sistemen irudia.</i>	23
4.1	<i>Proposatutako arkitekturaren diagrama.</i>	31
4.2	<i>Erabilitako metodoaren mapa kontzeptuala</i>	34
4.3	<i>Caching sistemaren klase diagrama</i>	39
4.4	<i>Irteerako Delivery Delay-ren Probabilitate dentsitate funtzioaren adibide bat.</i>	41
4.5	<i>Simulatzailerearen klase diagrama</i>	43
4.6	<i>Zipf banaketa <math>\alpha</math> ezberdinekin</i>	45
4.7	<i>Emaitzak alpha ezberdinetarako.</i>	48
4.8	<i>Emaitzak cache tamaina ezberdinetarako.</i>	50
4.9	<i>Emaitzak cache tamaina ezberdinetarako.</i>	52



---

## Taulen aurkibidea

---

2.1	Emangarriak. . . . .	14
4.1	Sailkatzaile bayesiarraren atributuak . . . . .	36
4.2	Simulazioen parametro konstanteak . . . . .	47



# 1. KAPITULUA

---

## Sarrera

---

2020rako, 5G public private partership<sup>1</sup> programak (5g-ppp), egungo komunikazio sareak baino aberats eta konplexuagoak espero ditu. Beraien aurreikuspenen arabera, sarea era eraginkor, mugikor eta malguan elementu hauek konektatzeko gai izango da: jendea, prozesuak, konputazio zentroak, edukiak, jakituria, informazioa, ondasunak, etab. Etorkizunak ekarriko dituen berrikuntzetarako ere prest egon beharko dute, hala nola, sentsore konektatuak, ibilgailu konektatuak, metrika adimentsuak edota etxeko aparatu adimentsuak.

Bideoetan oinarritutako aplikazioen ospea gero eta handiagoa da, ondorioz, gero eta banda zabalera handiagoa eta datuen erabilera hobea eskatuko dute. Errealitate areagotua bezalako teknologiak modan jartzen ari dira, baina gaur egun latentziarekin arazoak dituzte. Beste aldetik, sentsore, eragile eta gauzen internetak (Internet of Things) sarean era eraginkorrean aplikatzeko, fluxu baxuko datuen trafikoa eta seinalizazioan hobetu beharko dute batez ere.

5G abiadura handiko sare berriekin, makinei urruneko datuetarako atzipena datu lokalen atzipenaren antzeko abiaduran egitea ahalbidetuko da.

Gaur egungo sare mugikorretan, cache eta konputazio ahalmenak handiak dira, bai *Base station*-etan (BS) edo mugikorretan. Beraz, funtsezkoa da ahalmen hauek eraginkorri erabiltzea eskaintzen diren eduki guztietarako. Hurbilketa interesgarri bat, zerbitzuen hornitzaileen haririk gabeko mugikor sareen ertzetan, BS eta Radio Access Network-etan, edukiak cacheatzea izan daiteke.

---

<sup>1</sup><https://5g-ppp.eu/>

Beraz, erabiltzaile batek eskaera bat egitean, zerbitzarirainoko bidea egin beharrean, BBU edo ertzetako cachetik lortuko du edukia, beti ere hau aurretik cacheatua izan bada. E-maila, logikoki, eskaera hori azkarrago zerbitzatuko dela izango da. Gainera, cachea eraginkorra bada, interneteko sarearen karga gutxituko da. Azken hau oso egokia da etorkizuneko sareetarako, aurretik esan bezala, etorkizunean eskakizunak oso handiak izango direlako.

Lan honetan proposatuko dena sare hauen konputazio eta biltegiatze ahalmena baliatuz, cache sistema eraginkor eta adimentsu bat sortzea izango da, bayesiar sailkatzaileak eta algoritmo jaleak erabiliz.

## 1.1 Testuingurua

### 1.1.1 Vicomtech-IK4 zentroa

Vicomtech-IK4<sup>2</sup> Donostiako Parke Teknologikoan kokatuta dagoen ikerketa aplikatuko zentroa da, 2001ean sortua, eta Computer Graphics, Visual Computing eta Multimedia teknologietan espezializatuta dago.

Vicomtech-IK4ren helburua da enpresen eta erakundeen berrikuntza beharrei erantzutea. Horretarako:

- Ikerketa aplikatuan dihardu, eta ikus-elkarrekintzako eta komunikazioko multimedia-teknologiak garatzen ditu.
- Industriekin, unibertsitateekin eta beste Zentro Teknologiko batzuekin lankidetzak estua du, horien osagarri delarik.
- Ikerlarien mugikortasuna eta prestakuntza sustatzen ditu.

Bi aliantza estrategikotan parte hartzen du:

- IK4 bertako eremuan: Euskal Herriko puntako 9 Zentro Teknologikoz osatua dago.
- GraphicsMedia.net nazioarteko eremuan: Computer Graphics eta Multimedia teknologiek lotutako ikerketa aplikatuan diharduten nazioarteko zentroyen sarea.

---

<sup>2</sup><http://www.vicomtech.org/>

## Historia

Vicomtech-IK4k 2001eko apirilean hasi zuen jarduera, Alemaniako Zentro Teknologiko sare baten eta hemengo administrazioen ekimenaren ondorioz. Ikus-konputazioko teknologietan espezializatuta dagoen zentro-sare honek Alemaniatik kanpo antzeko zentro bat sortzeko asmoa adierazi zuen. Eusko Jaurlaritzak ekimena bultzatu zuen bertako beste administrazioekin batera eta horrela, Donostiako Parke Teknologikoan Vicomtech-IK4 sortu zuten.



### **1.1 Irudia:** *Vicomtech-IK4 zentroaren logoa.*

Hastapenetan, zentroak 6 langile zituen; baina pixkanaka haziz joan da, gaur egun ehun langile inguru dituelarik. Fakturazioa ere etengabeen igo da urtez urte, zentroa osatzen duten pertsonen irmotasunari eta konpromisoari esker, neurri handi batean.

2006an, zentroa IK4 Aliantza Teknologikoaren barruan sartu zen, puntako 9 euskal zentroz osatutakoa. Nazioartean, 2011n Graphicsmedia.net fundazioan sartu zen, Computer Graphics eta Multimedia teknologietako ikerketa aplikatua diharduten zentro ospetsuez osatzen den nazioarteko sarean, hain zuzen ere. Horrek profil aktiboa eta estrategikoa ematen dio, ikerketa-jarduera nazioartekotzeari dagokionez, eta ikertzaileen mugikortasuna eta ezagutzen trukea errazten ditu.

## Digital Media Saila

Informazio gizartean gailu mugikorretarako joera argia nabarmendu da azken urteotan, eta kanalak, komunikabideak eta gailuak bateratuz doaz era horretan erabiltzailean esperientzia aberasten eta hobetzen dutelarik. Guzti horretarako, multimedia-fluxuak eta elkarrekintza-gaitasunak erraztasunez integratzen dituzten irtenbide teknologikoak behar dira. Vicomtech-IK4k "Multimedia-Etxea" egia bihurtzeko ezagutza eta teknologia-oinarri sendoak ditu, erabiltzailean oinarritutako edukiak eta zerbitzuak eskainiz.

Bestetik, telebista digitala ere multimedia-zerbitzuen bateratze prozesu horren parte bihurtzen ari da, igorpenak eta banda zabaleko komunikazioa modu hibridoan konbinatzen dituzten hurbilketa teknologikoen bidez, erabiltzailearen esperientzia hobetze aldera.

Sail hau ere Big Data eremua ikertzen ari da, edukiera memoria, zenbatze eta banda-zabalerari dagokionean cloud computing teknologien bidez sistemen eskalagarritasuna bermatzeko. Era honetan, azpiegiturak lan kargari dinamikoki egokitzen zaizkiolarik. Edukien analisisien, indexazio eta kudeaketa paradigman aldaketa sakona dakar honek. Vicomtech-IK4k konputazioak hodeian dituen ezaugarrietara egokitzen diren analisi teknika berrien garapenean ikertzen dihardu, metodoen sailkapen eta datuen bistaratze modu berriak eskainiz.

Sail honetan eta aipatutako azken eremu hontan kokatua dago egin beharreko lana.

### 1.1.2 CogNet Europar Proiektua

Txosten honetan azaldutako proiektua, lehen adierazi bezala, CogNet<sup>3</sup> europar proiektuaren barnean dago. Ondorengo lerroetan honen sarrera bat, helburu nagusiak, kideak eta datu gehigarriak azalduko dira.

#### Sarrera

CogNet, Europar batasunak eta 5G elkarteak jaurtitako 19 proiektuetako bat da 5g-ppp-aren lehen fasea inplementatzeko.

CogNet-ek sareen eta hauen aplikazioen monitorizazio eta kudeaketa beste maila altuago eta adimentsuago batera eramateko soluzioak garatzeko sortu da, eragiketen eraginkortasuna hobetuz, horrela 5G sarrentzako eskakizunei aurre egitea errazago izan dadin.

Proiektuak datu bilketa, ikasketa automatikoa, datu analisisia eta sareen kudeaketa automatikoen ataletan ikertuko du. Honetaz gain, sare topologia handiago eta dinamikoagoak sortzea du helburutzat, 5G sareetan beharrezkoa izango dena, horrela erabiltzaileen zerbitzuaren kalitatea (QoS) hobetuko da, eta eragiketen kostua jeitsi egingo da sareko nodoen, loturen eta funtzio birtualizatuen eraginkortasuna hobetuz.

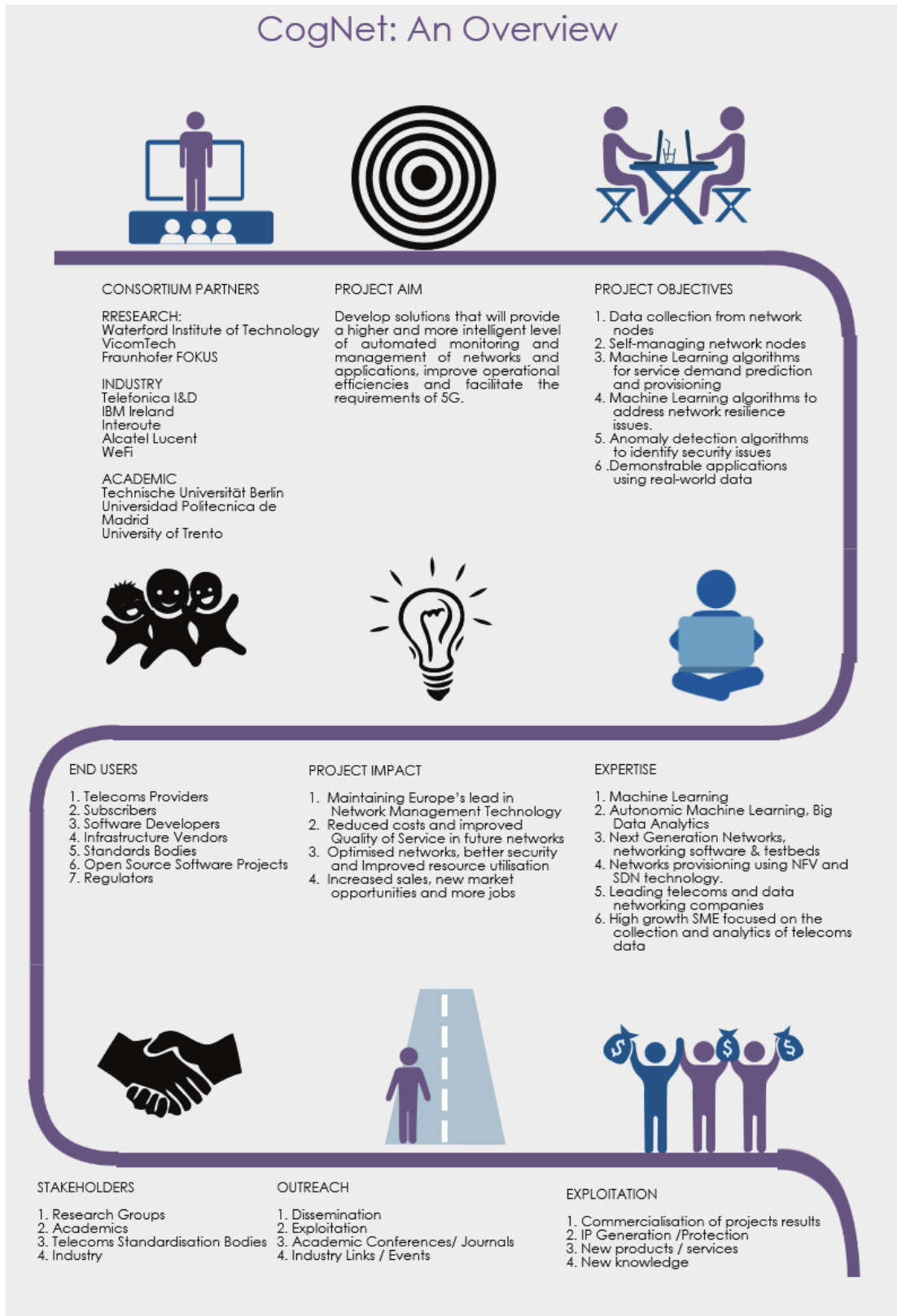
#### Helburu Nagusiak

1. Sareko nodoen datu bilketa, aurreprozesaketa, eta sailkapena, datu irregular eta garrantzitsuenak identifikatzeko, ohiko datuak filtratzen dituen bitartean. Pausu ga-

---

<sup>3</sup><http://www.cognet.5g-ppp.eu/>





1.2 Irudia: CogNet Proiektuaren balio proposamena.

rrantzitsua da sare eskalagarrien kudeaketan, prozesatu behar diren datuak asko gutxitzen direlako.

2. Sare autonomoen hastapenen lanean ari diren bitartean, ikertu eta garatu, dagoeneko existitzen diren sareen kudeaketa politika *framework*-ak erabiliz, nodoen kudeaketa autonomoa bermatuko duen sistema bat, bere datuetan oinarrituta.
3. *Machine Learning* algoritmoak aplikatzea zerbitzu eskaeren aurreikuspenen sistema bat garratzeko, sarearen tamaina eta baliabideetan aldaketak bideratzeko, birtualizazioa erabiliz, aurreikusitako eskaerei zerbitzua emateko adostutako parametroekin, hala nola, kokalekua, denbora, eta zerbitzu espezifikoaren eskaerak erabiltzaile edo erabiltzaile talde konkretu batek egina. Makina birtualen eta sarearen portara optimizatzea lortu behar da, kostu eta energia baliabideak minimizatzen diren bitartean.
4. *Machine Learning* algoritmoak erabiltzea sareari iraunkortasunaz doitzeko. Garatu behar diren atazen artean, gainbegiraturako algoritmoak erabiltzea sarearen errore eta baldintzak identifikatzeko, kongestioak bezalakoak. Behin identifikatuta, automatikoki arazo hauek arintzeko ekintzak gauzatu.
5. Anomalien detekziorako algoritmoak erabiltzea segurtasun arrisku garrantzitsuak identifikatzeko.
6. Aplikazio erakusgarriak garatzea, mundu errealeko datuak lortuz, gaur egungo 4G sareko nodoen datuak erabiliz, horrela proiektuaren berrikuntzak frogatzeko, eta CogNet-en esplotazio potentziala erakusteko.

#### Proiektuko kideak

- Waterford Institute of Technology's Telecommunications Software and Systems Group (TSSG) (Koordinatzailea)
- Telefonica I+D
- IBM
- Vicomtech-IK4
- Interoute

- Fraunhofer FOKUS
- Technische Universität Berlin
- Universidad Politécnica de Madrid
- Alcatel-Lucent Israel
- Università di Trento
- Orange

#### Bestelako Datuak

CogNet H2020 5GPPP proiektua da, Europar batasunak jaurtia ondorengo ICT sailaren barnean: *ICT-14-2014 Advanced 5G Network Infrastructure for the Future Internet*

- **Hasiera data:** 01/07/2015
- **Bukaera data:** 31/12/2017
- **Kostua:** 5.972.820 €
- **Finantzaketa:** 5,972,820 €
- **Estimatutako esfortzua:** 672.5 Pertsona Hilabete
- **Proiektuaren identifikatzailea:** H2020-671625



## 2. KAPITULUA

---

### Kudeaketa

---

Kapitulu honetan, proiektuaren kudeaketarekin erlazionatutako atalak aztertuko dira. Helburuak eta motibazioak definitu ondoren, plangintza zehaztuko da, baita lan metodologia ere.

#### 2.1 Helburuak eta motibazioa

Proiektu honen helburu nagusia 5G saretarako kudeaketa automatikoa duen sistema bat sortzea izango da *machine learning* teknikak erabiliz. Horretarako lehenik Ikerkuntza, garapen eta berrikuntza (I+G+b) lanak egin beharko dira.

Aparteko helburu bezala, I+G+b proiektuak dituzten difusioa egitea da, hau da, artikulua idaztea eta artikulua onartua bada konferentziaren batean, hitzaldi hori prestatu eta aurkeztea.

Orain arte unibertsitatean egindako lan guztiak, lan munduan gertatzen denaren prestakuntza bezala burutu dira. Hala ere, lanean hasi arte, ezinezkoa da errealitatean gertatzen denaz jabetzea.

Horregatik, proiektu honen motibaziorik garrantzitsuena unibertsitate mundutik atera eta lan munduan, batez ere teknologi zentro batean buru belarri sartzea eta errealitatean gertatzen denari buruz ikastea da.

Azken honen bidez, I+G+b proiektuetan lan egiten ikastea, eta proiektu handi baten parte hartzea, kasu honetan europar proiektu batean.

Azkenik, lan ingurune berri batera eta lan talde handietan lan egitera moldatzea ere motibazioetako bat izango litzateke.

## 2.2 Plangintza

Nahiz eta plangintza orientagarria den, ezbehar asko gerta daitezkeelako proiektu baten bizi zikloan zehar, lanaren planifikazio egoki bat egiteak buruhauste asko kenduko dizkigu.

### 2.2.1 Atazen identifikazioa

Oso garrantzitsua da egin beharreko atazak ongi identifikatzea, honek ondoren egin beharreko lana erraztuko baitigu.

- *Proiektuaren kudeaketa:*
  - *Plangintza:* Proiektuaren plangintza eta estimazioak egitea.
  - *Jarraipena eta kontrola:* Proiektuan ematen diren pausoen jarraipena eraman eta hauen desbideraketen arrazoiei buruz hausnartu eta soluzioak hartzea.
- *Artearen egoeraren analisia:* I+G+b proiektuetan oso garrantzia handia du artearen egoeraren analisi egoki bat egiteak. Gure proiektua aurrera eramateko beharrezko teknologia berrienak ezagutu, eta hauek hobetzeko bideren bat aurkitzea da honen xedea.
- *Beharrezko tresnen analisia:* Proiektua garatu behar izateko, beharrezkoak diren teknologia berriei buruzko hausnarketa eta analisia egitea.
- *Bilerak proiektuko kideekin:* Proiektuko kideekin komunikazioa garrantzitsua izan-go da maila hontako proiektuetan. Hemen nire partaidetza baxua izango da, nire zentroko proiektu buruak izango dituelako bilera gehienak.
- *Garapen teknikoa:*
  - *Analisia:* Gure aplikazioak izan beharko dituen funtzionalitateak aztertuko dira.

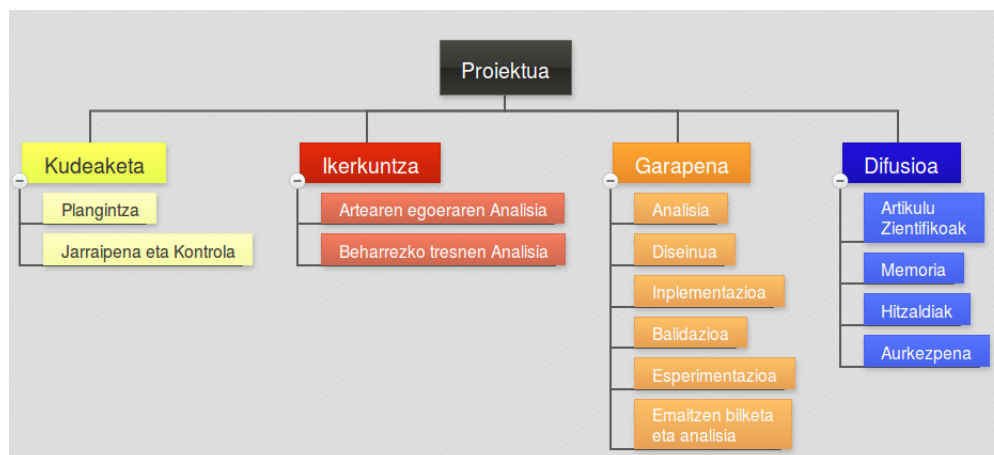
- *Diseinua*: Aplikazioak izan beharreko funtzionalitateen arabera, honen diseinu bat egingo da.
  - *Inplementazioa*: Aurretik diseinatutako funtzionalitatea implementatuko da.
  - *Balidazioa*: Sistema gure proiekturako eskakizunekin bat datorrela egiaztatzea.
  - *Esperimentazioa*: Probak egingo dira egon daitezkeen erroreak identifikatzeko eta baita gure aplikazioaren parametroak optimizatzeko.
  - *Emaitzen bilketa eta analisisa*: Behin gure aplikazioaren parametro ezberdinekin probak egin ondoren, emaitzak interpretatu, konparatu eta analizatu egingo dira sistema optimizatzeko helburuarekin.
- *Difusioa*:
    - *Artikulu zientifikoaren idazketa*: Teknologia berri bat sortzen den heinean, hau hedatzea artikulu zientifiko baten bidez.
    - *Memoriaren idazketa*: Zati hau garrantzitsua da, izan ere bertan islatuko dira proiektuan zehar egin den lan guztia.
    - *Hitzaldiak*: Gure proiektua konferentzia batean azaltzea, eta honen prestakuntza.
    - *Proiektuaren defentsa*: Proiektuaren amaieran unibertsitateko tribunalaren aurrean egingo den aurkezpena antolatu eta prestatuko da.

## 2.2.2 Lanaren Deskonposaketa Egitura (LDE)

2.1 irudian azalduko da proiektuaren lanaren deskonposaketa egitura.

## 2.2.3 Planifikatutako egutegia

Atal honetan, gure plangintzan planifikatutako datak Gantt diagrama modura azalduko dira. 2.2 irudian aurki daiteke diagrama hau.



**2.1 Irudia:** *Proiektuko Lanaren Deskonposaketa Egitura*

## 2.3 Lan Metodologia

Atal honetan, proiektua garatzeko metodologia azalduko da. Lehenik kudeaketa eta antolaketan zentratuko gara. Ondoren emangarriak definitu eta artxiboen kudeaketaz hitz egingo da. Azkenik arriskuak definitu eta hauen kudeaketaz hitz egingo dugu.

### 2.3.1 Kudeaketa eta Antolaketa

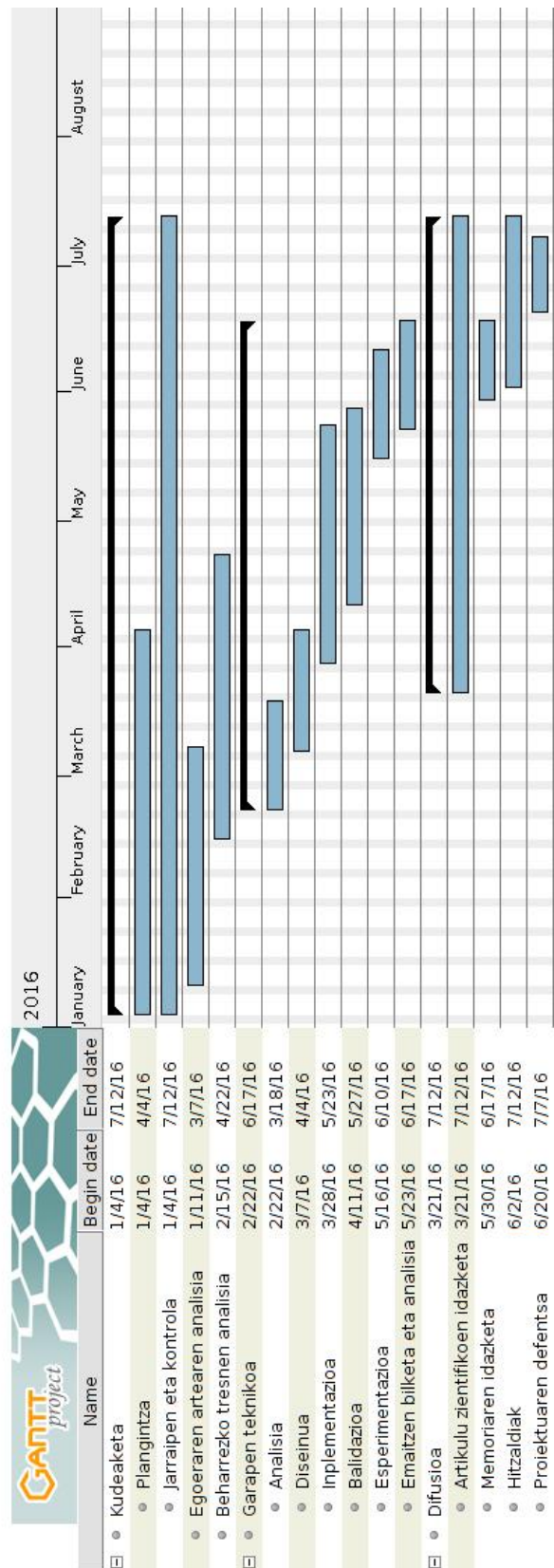
Proiektuaren kudeaketa aldetik ez da arazo handirik egongo, ni proiektuaren langile bakarra izateko pentsaturik dagoelako. Baina, lan hau CogNet-en barruan dagoenez, Vicomtech-IK4-ko CogNet proiektuaren barnean dauden beste kideekin bilerak egin beharko dira proiektuaren irismena zehazterako eta garatzerako orduan, CogNet-en eskakizunak betetzen ari naizela egiaztatzeko.

Nire orduen jarraipen eta kontrola egiteko, Vicomtech-IK4-k kudeaketa sistema bat du, *Ekhi* deiturikoa, hor lan orduak sartu eta Zentro teknologikoan egin daitezkeen ekintzak kudeatzeko, adibidez, bidaiak prestatu, artikuluko zientifikoaren kudeaketa egin, bilera gela bat erreserbatu, etab.

### 2.3.2 Emangarriak

Proiektuak hainbat emangarri sortu beharko ditu, beraz horiek definitzea eta hauen emate datak definitzea garrantzitsua da.





2.2 Irudia: Proiektuaren Gantt diagrama

Emangarria	Deskribapena	Formatua	Entrega-data
Kodea	Sortutako kode guztia	Konprimitua (.zip, .rar, .tar.gz)	Ekainak 24
Gidaliburua	Kodearen espezifikazio teknikoa	PDF	Ekainak 24
Memoria	Proiektuaren memoria	PDF	Ekainak 24
Aurkezpena	Proiektuaren defentsaren aurkezpena	PDF	Uztailak 8,11,12
Artikulua	Idatzitako artikulua zientifikoa	PDF	Maiatzak 13
Hitzaldia	Prestatutako hitzaldi zientifikoa	PDF	Ekainak 27

### 2.1 Taula: Emangarriak.

3 emangarri mota nagusi identifikatu dira, teknikoak, dokumentazioa, eta difusiozkoak.

2.1 taulan ikus daitezke emangarriak, deskribapen motz bat, emateko formatua, eta entrega-data.

#### 2.3.3 Artxiboen kudeaketa

Ikasleak bere lanpostuko ordenagailuan kopia lokal bat gordeko du bere karpeta pertsonalean. Direktorio hau bi atal nagusitan banatuko da, batetik kodea, eta beste aldetik dokumentazioarekin erlazionatuta dagoen guztia.

Kodea, Vicomtech-IK4-ko sailak duen GitHub kontu batean ere egongo da eguneratua. Beraz, Git bertsio kontrolerako softwarea ere erabiliko dugu. Zerbitzu honek ere, lantokitik kanpo lan egiteko aukera emango digu.

Memoriari dagokionez, ShareLateX webgunean idatziko da, eta egunero karpeta pertsonalera jaitziko dugu azken bertsioa. ShareLateX webguneak ere bertsio kontrol sistema bat du. Sistema honek ere, memoria partekatzeko aukera du, bere doako bertsioan, pertsona batekin bakarrik. Beraz tutoreek ere memoria eskura izango dute nahi dutenean.

## 2.4 Arriskuen Kudeaketa

Arazoen aurrean erabaki azkar eta egokiak hartzeko, garrantzitsua izaten da arriskuen kudeaketa ona egitea, garapena geldirik ez geratzeko. Beraz, lehenengo atala, arriskuak identifikatzea izango litzateke, ondoren erantzunak ondo prestatzeko.

Atal honetan proiektuan zehar izan daitezkeen arriskuak identifikatuko ditugu, baita hauek nola konpondu adieraziko dugu.

- *Gaixotzeak*: Proiektuan zehar baliteke partaideren bat gaixotzea. Kasu hauetan ezin da asko egin, ahal den moduan kontaktuan jartzeko mekanismoak erabili daitezke arazo larri baten aurrean, adibidez, kontaktu telefonikoa, telekonferentzia edo posta elektroniko bidez.
- *Arazo teknikoak*: Atal honetan, software edo hardwarearekin egon daitezkeen arrikuak identifikatu ditugu. Arazo baten aurrean, Vicomtech-IK4-ko sistemen sailera jo eta haiek konponduko dute.
- *Informazio galera*: Sistema ezberdinetan sor daitezkeen arazoak direla eta, informazio galera gerta daiteke. 2.3.3 puntuan adierazi bezala, bi modutara gordeko dira fitxategi guztiak, hodeian, bertsioen kontrola dutela, eta kopia lokal batean.
- *Ikaslearen lanaren eta ikasketen bateraezintasunak*: Kasu hontan, proiektua gelditzen bada, proiektua berriz hartzean, sartutako ordu horiek 'ordu extra' bezala sartuko dira.
- *Tutorearen lan karga handia*: Vicomtech-IK4-n udaran praktiketan ibili ondoren, zentroko tutorearen lan karga oso handia izaten dela ohartu naiz. Beraz, lan karga handia duen momentuetan zaila izaten da berarekin eseri eta proiektuko gauzetaz hitz egitea. Hau ekiditeko, bere Google Calendar-a izango dut eskura, bilerak, emate datak edo bidaiak noiz izango dituen aurrez jakiteko, hauei aurre hartu eta lasai hitz egiteko aukera izateko.

## 2.5 Baliabideaen Kudeaketa

### 2.5.1 Giza Baliabideak

- *Basilio Sierra*: Gainbegirale akademikoa, EHU-ko irakaslea.
- *Marco Quartulli*: Vicomtech-IK4-ko gainbegiralea. Digital Media saileko ikerlari seniorra.
- Vicomtech-IK4 Digital Media saileko ikerlariak.

### 2.5.2 Baliabide materialak

- Bulego materiala.

- Bilera gelak.
- Liburuak.
- Argitalpen zientifikoak.
- Interneteko konexioa duen ordenagailua
- Softwarea
  - Linux
  - Mozilla Firefox
  - Open Office
  - Latex
  - Gantt Project<sup>1</sup>
  - FreeMind<sup>2</sup>
  - Spyder<sup>3</sup>
  - iPython Notebook<sup>4</sup>
  - Sublime<sup>5</sup>
  - yEd<sup>6</sup>

---

<sup>1</sup><https://www.ganttproject.biz/>

<sup>2</sup>[http://freemind.sourceforge.net/wiki/index.php/Main\\_Page](http://freemind.sourceforge.net/wiki/index.php/Main_Page)

<sup>3</sup><https://pypi.python.org/pypi/spyder>

<sup>4</sup><https://ipython.org/notebook.html>

<sup>5</sup><https://www.sublimetext.com/>

<sup>6</sup><https://www.yworks.com/products/yed>

## 3. KAPITULUA

---

### Ikerketa

---

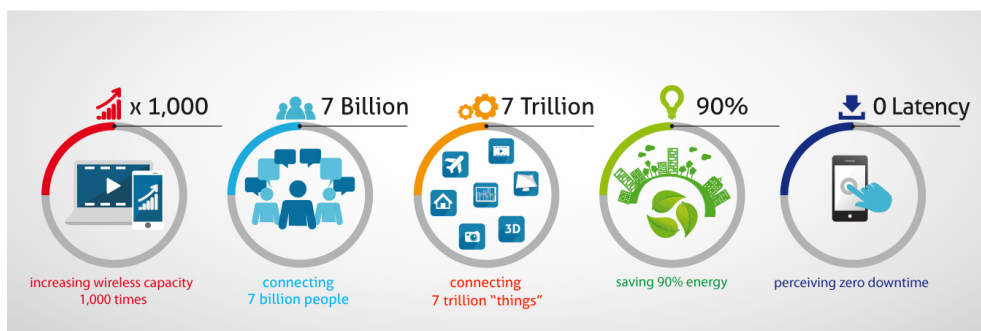
Kapitulu honetan, egindako ikerketen berri emango da. CogNet proiektuak zituen eskakizunen aurrean, artearen egoeraren erretratu txiki bat egingo da, gaian pixka bat sartzeko erabiliko dena. Bigarrenik, behin artearen egoera ikusita, eta honen arabera norabide bat aukeratuta, garatzeko behar diren tresnen analisi eta aukeraketa egin da.

### 3.1 Artearen Egoera

#### 3.1.1 5G Sareak

5G sareekin batera etorriko da benetako gauzen interneta, haririk gabe konektatutako 7 trilioi gailu espero dira, eta gaur egun dagoen banda zabalera 1.000 aldiz hobetzea espero da. Egungo 4G teknologia, generazio honekin posibleak diren limiteetara hurbiltzen ari da. Hau aurrera eramateko, garrantzitsua izango da sarea oso optimizatua egotea bere erabileraren maximoa eman dezan, horrela datu transmisio abiadura handiago bat egon dadin, zerbitzuaren kalitatea igoz.

Lehen aipatu bezala, gailu konektatu asko daudenez, garrantzitsua izango da baita ere sarearen kudeaketa autonomoa egitea, batez ere antolakuntza, konfigurazio, segurtasun eta optimizazio ataletan. Birtualizazioak paper garrantzitsu bat izango du, bere baliabideak dinamikoki aldatzeko beharrak izango baititu. Hortarako giltza *Network Function Virtualisation* (NFV) izango da, sareko nodoen, funtzioen eta loturen birtualizazioan oinarritzen



**3.1 Irudia:** 5GPP-aren etorkizuneko estimazioak.

dena, beraz eskalagarria izan daitekeen beharren arabera, estimatutako eskaera maximo batean oinarritutako sare bat eraikiz baino.

#### Cloud Radio Access Networks (C-RAN)

C-RAN sareak, etorkizuneko sare zelularretarako proposatutako arkitektura bat da. C-RAN sareak zentralizatuak dira, Radio access network-etan *cloud computing*-a errealitate bihurtzen du. 2G, 3G, 4G eta 5G sareetan lan egiteko gai da.

RAN sareak, Antena zelularretara konektatzen diren oinarritzko estazio independenteez osatuta dago. Gailu hauek, seinaleak prozesatu eta sarearen nukleora bidaltzen ditu. Normalean Base Band Unit (BBU) izena hartzen dute. Sare hauetarako konektibitatearen eskaera asko igoenez urte hauetan, mugikorren operadoreak, eskari hauek eraginkorki zerbitzatzeko kostu ahalik eta txikienean, RAN sareen zati batzuk zentralizatzeari ekin diote, Cloud-RAN kontzeptua sortuz.

Normalean, BBU eta Radio Remote Head (RRH) antenak zuntz optikoz konektatuak izaten dira, eta C-RAN arkitektura batzuetan, hainbat BBU-k informazioa partekatu dezakete, sareak eskaini dezakeen baliabideen erabilera eraginkorragoa eginez.

C-RAN arkitekturek honako ezaugarriak dituzte, beste sare zelularrekin alderatuz:

- Base Band Unit zentralizatu batera, ehun eta milaka RRH antena konektaturik izatea ahalbidetzen du. Hauen arteko distantzia maximoa 20km-koa da, zuntz optikoz konektaturiko 4G sare bat izanik. 3G eta 2G sareetarako oraindik distantzia handiagoa da, 40km-tik 80km arte.
- Irrati teknologia kolaboratiboetarako plataforma oso egokia da. Edozein BBU-k

beste BBU batekin datuak elkarbanatu ditzake banda zabalera oso handiarekin eta latentzia oso txikiarekin.

- Denbora errealeko birtualizazioa posible egiten du. Honek posible egiten du baliabideak eraginkorki esleitzea BBU-ei, dagoen eskaeraren arabera.

### Software Defined Networks (SDN)

Software bidez definitutako sareak izango litzateke euskarazko itzulpena. Sareen arkitektura berri bat da. Honako hauek dira SDN-ren ezaugarriak:

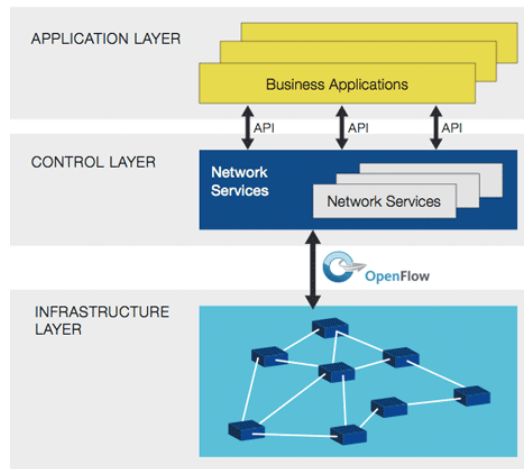
- Dinamikoa
- Kudeagarria
- Ekonomikoki jasagarria
- Moldagarria

Ezaugarri hauek, SDN sareak idealak bihurtzen dituzte banda zabalera handia behar dituzten eta dinamikoak diren egungo aplikazioentzat. Arkitektura honek sarearen kontrolaren eta birbidalketaren funtzioen arteko lotura hausten du, sarearen kontrola zuzenean programaria bihurtzean, azpiegiturak aplikazio eta sare zerbitzuetatik urrunduz. Laburbilduz, SDN sareek, kontrol eginkizunak sareko bideragailuetatik kanpo dagoen kontrolatzaile izeneko entitatera kanporatzen ditu.

SDN sare hornitzaileak hainbat arkitektura eraginkor eskaintzen dituzte, sarearen kontrola datuen kontroletik urruntzeko. Egitura honakoa da: SDN kontrolatzaile bat, Hegoalderako API bat (Southbound API) eta Iparralderako API bat (Northbound API).

**SDN Kontrolatzailea** SDN kontrolatzaile sarearen garun bezala aritzen da. Hau da, azpiegiturako switch/router-etara Southbound API-aren bitartez, eta aplikazio eta negozio logikako geruzetara Northbound API-aren bitartez informazioa bidaltzen du. Gaur egun bi interfaze nagusi erabiltzen dira: OpenFlow eta OVSDB(Open Virtual Switch DataBase).

SDN kontrolatzaileak modu errazean konektatu eta deskonektatu daitezkeen modueletan oinarritzen da. Modulu hauek sareen ataza ezberdinak egin ditzakete, adibidez sareko aparatu guztien inbentarioa, hauen gaitasunak bildu, sareko estatistikak taldekatu, etab.



**3.2 Irudia:** SDN sareen geruzen errepresentazioa.

Norhtbound eta Southbound API-ak Aurreko azpiatalean esan bezala, interfaze hauek SDN kontrolatzailea azpiegitura eta aplikazio geruzekin konektatzeko balio dute.

Southbound API-ek sarearen kontrola errazten dute, honi esker kontroladoreak dinamiko-ki aldaketak egin ditzake, denbora errealean izan daitekeen eskarien eta beharren arabera. OpenFlow, *Open Networking Foundation* (ONF)-k sortua, egin den lehena, eta ezagunena da baita ere.

Beste aldetik, NorthBound API-ek, berrikuntza errazteko eta SDN-en programagarritasunari esker, erakundeei sarearen antolakuntza eta automatizazioa eskaintzen zaie bere aplikazioen beharren arabera.

### Network Functions Virtualization

Sareko osagai garrantzitsuak birtualizatzean datza, hala nola *firewall*-ak, *router*-ak, konmutagailuak, biltegitratze sistemak, karga orekatzaileak, etab. Beraz, ez da beharrezkoa sareko tresneria fisikoak edukitzea, eta beren funtzio berdinak egin ahal izatea du helburutzat, softwarean oinarritutako sareko funtzio eta zerbitzuak erabiliz. Guzti hau zerbitzari arruntek egin dezakete, era zentralizatu edo banatu batean, birtualizazio kapa baten azpian.

Hardware hau normalean prestatutako bat edo hainbat zerbitzariz osatua dago, gaitasun fisiko nahikoa izan behar dute (Intel eta ARM Holdings<sup>2</sup> prozesadore espezifikikoak garatzen hasi dira) hainbat makina birtual jasateko. Makina birtual hauek lehen aipatutako



tresneria fisikoa ordezkaturako dituzte, beraz, beraien funtzio eta adimena, zerbitzari arrunt batek exekutaturako duen makina birtual batean egongo dira.

Era hontara, birtualki sare arkitektura oso bat bakarrik makina fisiko bakarrean izan dezakegu. Sare komunikabideetan aurrerapen handia da teknologia hau, kostuak asko murrizten direlako, ez duzulako sareko hainbat tresneria fisiko erosi behar, zerbitzari arrunt bat edo gehiagorekin nahikoa izango litzateke. Hontaz aparte, sarearen kudeaketa hobe eta sinpletasuna konfiguratzeko orduan eskaintzen du.

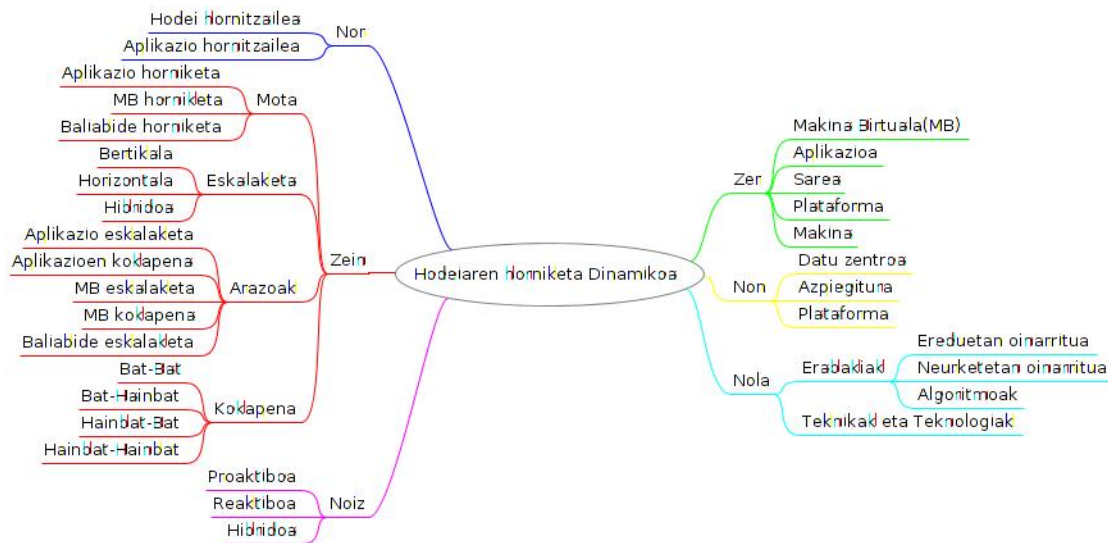
Hasiera batean NFV operadoreen sareentzako planteatu zen, baina gaur egun edozein *datacenter*-erako baliagarria izan daiteke. Izan ere, badira adituak enpresa mailan gehiago erabiliko direla diotenak.

### 3.1.2 Hodeiaren horniketaren Sailkapena

Atal honetan, hodeien sistemen horniketa dinamikoaren arazoa ikertzen da. Bide asko har daitezke, eta [SD14] artikulutik ateratako 3.3 irudian bide horiek sailkatuak azaltzen dira, eta ondoren gehiago zehaztuko dira.

Egia esan, nahiz eta lanak norabide hau ez hartu, memorian sartzea pentsatu da, arazo honen ikerketan denbora asko sartu delako, gure lanaren norabidea zehaztu aurretik.

- **Nor:** Horniketa egiten duen entitatea izango litzateke. Hodeiaren horniketetan, hiru entitate ezberdintzen dira, Hodei hornitzaileak (HH), Aplikazio hornitzaileak (AH) eta erabiltzaileak. Normalean hodei hornitzaileak Azpiegiturak zerbitzu bezala (IaaS) hornitzen dituzte, hardwarearen gainean dabilzaten makina birtualak bezala. Aplikazio hornitzaileak izaten dira IaaS-en bezeroak, Makina birtualak sortu eta istantziatu dezaketenak, ingurunea aukeratu, eta softwarea instalatzen dutenak.
- **Zein:** Zein horniketa mota, eskalaketa, arazo eta kokapen politika existitzen diren. Era sinpleago batean ikusita, aplikazio, makina birtual eta baliabide horniketa bezala sailka daitezke baita ere.
- **Noiz:** Noiz egiten den horniketaren erabakia. Bi mota nagusi existitzen dira, erreaktiboa eta proaktiboa. Lehenengoak, eskaera baten aurrean erreakzionatzen du horniketa egiteko, edo eskaeretan aldaketak daudenean ere. Bigarrena berriz aurreikuspenez baliatzen da horniketak burutzeko. Hau da, aurreikuspen bat egiten du, eta aldaketarik balego, aldaketa horietarako prestatzen hasiko litzake. Azken modu bat ere lantzen hasi da, hibridoa, hau da, bi azken hauek nahasten dituenak.



**3.3 Irudia:** Hodeiaren horniketa dinamikoaren mapa.

- **Zer:** Hainbat baliabide hornitzen dira hodeietan zerbitzu bezala:
  1. Makina birtualak eta honekin erlazionaturako baliabideak hornitzen dira IaaS bitartez.
  2. Garapen plataformak eskaintzen dira Plataforma zerbitzu bezalako (PaaS) horniketekin.
  3. Softwarea eskuragarri dago Softwarea zerbitzu bezalako (SaaS) horniketekin.
- **Non:** Erabakiak non hartzen ari diren. Hodei hornitzaileak azpiegitura mailan hartzen baditu erabakiak, maila global batean hartuko litzuke, baliabideen erabilera optimizatzeko. Beste aukera batzuk *datacenter*-etan edota plataformetan burutzea izango litzateke.
- **Nola:** Atal honetan arazoak nola ebatziko diren aztertzen da, algoritmo, teknika eta teknologia ezberdinak erabiliz.

### 3.1.3 Cache Algoritmoak

Gaur egun, cache algoritmo asko proposatu dira. Hurrengo atalean, nagusienak edo erabilienak aipatuko dira. Gure proposamena bi zatitan banatua dagoenez, hemen ere bi zatien

algoritmo nagusienak azalduko dira.

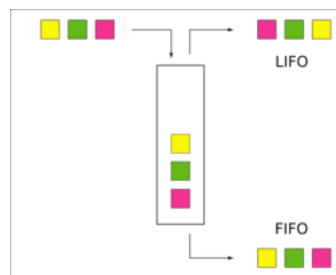
Erabakitze algoritmoak

Atal honetan, zein objektu cacheatua izan behar den zehazteko algoritmoak argituko dira. Ondoren probetan hauetako batzuk erabiliko dira.

**Ausazkoa** Ausaz objektu bat aukeratu eta hori ezabatuko du cacheatu behar den objektua cachean sartzeko. Ez da oso erabilia.

**Least Recently Used (LRU)** Algoritmo hau oso sinplea da, objektu bat cacheatzerako orduan, bere tamaina cachean libratzeko, objektu zaharrenak ezabatuko dira, hau da, cachea FIFO ilara bat bezala irudikatu daiteke.

**Most Recently Used (MRU)** LRU algoritmoaren aurkakoa. Objektu berriena ezabatzen du. Hau da, LIFO ilara bat bezala irudikatu daiteke. Web objektuak cacheatzerako orduan ez da oso eraginkorra. Ordenagailu pertsonalen cachean gehiago erabili daiteke, askotan eskaera berdina egiten bada, eta hau askotan aldatzen bada eraginkorra izan daiteke, datu zaharrak ez erabiltzeko.



**3.4 Irudia:** *FIFO eta LIFO sistemen irudia.*

**Least Frequently Used (LFU)** Algoritmo honek objektuek izan duten eskaera kopuruan kontatzen du. Cacheatzen dituen objektuak gehien eskatuak direnak izaten dira. Berez algoritmo ona da, baina ez da urtarokotasunera ohitzen.

Greedy Dual Size (GDS) Algoritmo honek objektuen pare bat metadatu behar ditu, objektua lortzearen kostua, eta bere tamaina. Gainera, adinaren faktore bat gehitzen du. 1 Algoritmoan ikus daiteke algoritmo honen funtzionamendua.

```

while eskaera bakoitzeko do
  if obj cachean dago then
    |  $H(p) = L + c(p)/s(p)$ 
  else
    | while ez dago espaziorik cachean do
      |  $L = \min_{q \in M} H(q);$ 
      | Kendu cachetik  $q$  non  $H(q) = L$ 
    | end
    |  $p$  cachean sartu eta  $H(p) = L + c(p)/s(p);$ 
  end
end

```

**Algorithm 1:** GDS algoritmo orokorra

### Kokapen algoritmoak

Atal honetan, hainbat cache ditugunean, edukiak kokatzeko algoritmo nagusienak aipatuko dira. Aipatu beharrekoa da, hemen bakarrik kokapenaren atala azaltzen dela, hau da, cacheatzeko puntu denak, beti ere erabakitze algoritmoak edukia cacheatua izan behar dela dionean.

Leave Copy Everywhere (LCE) Algoritmo hau sinpleena da. Cacheak pasa behar dituen hierarkia maila guztietan cacheatzen da edukia. Kasu honetan era independientean erabakitzen dira cacheatuak izan behar diren, hau da, erabakitze algoritmoak hierarkia maila bakoitzean exekutatuko lirateke.

Kooperatiboak Cacheen arteko komunikazioa erabiltzen dute, cacheen edukia optimizatzeko. Hainbat ikerketa egin dira honen inguruan. Ohikoena problema hau ebazteko algoritmo jale eta bilaketa heuristikoak erabiltzean datza, problema hau NP-oso delako.

Badira guk analizatutako beste algoritmo batzuk literaturan. Guk erreferentzia modura hartu duguna [THP16], beste algoritmo kooperatibo bat proposatzen du. Simulazioa egi-terako orduan ere artikulu honen hainbat parametro hartu ditugu. 2016ko otsailan publikatua izan denez, balio erantsi bat bezala ikusi dugu, berria delako.

Beste lan batzuetan [BBD14][AD14][GSD<sup>+</sup>12], antena zelularrak cacheatzearen etekinak aztertu dituzte. [BBD14]-an zehazki, caching proaktibo bat proposatzen da BBU-ak interneterako duen loturaren kongestioa ekiditeko. [AD14]-n berriz, Konektatuak dauden Erabiltzaileen lehentasun profilak erabiltzen ditu, C-RAN-eko cache algoritmo erreaktibo eta proaktibo ezberdinetarako. [GSD<sup>+</sup>12]-ek *Femtocaching* izeneko terminoa sortzen du, *Femtocell* izeneko oinarrizko estazioen arteko loturak erabiltzen ditu caching banatu bat sortzeko.

Bestalde, [WCT<sup>+</sup>14] artikuluan, *Evolved Packet Core* eta RAN ertzetan cacheatzea proposatzen du, non lehenengoak izan dezakeen cache edukiera handiaz baliatu daitekeen, baina zerbitzatzeko denbora igo egiten da BBU eta EPC-ren arteko latentziagatik. [BGW10]-ean berriz, eduki ospetsuenen cache eskema bat sortzen du, non cache bakoitzean era independentean gordetzen diren edukiak, baina ertzetan daudenak hodei cachetik baztertzen dira.

## 3.2 Garapen tresnen analisia

Behin lan hau aurrera eramateko norabidea erabakita, behar ditugun tresnak analizatu behar ditugu. Hasteko, logikoki, gure proiektura gehien egokitzen den lengoaia hartu behar dugu, eta gure lanerako behar diren liburutegiak dituela egiaztatuz.

Lengoaia

Vicomtech-IK4-ko Digital Media sailan gehien erabiltzen den lengoaia *Python*<sup>1</sup> da, eta ni lengoiairekin familiarizatua nagoenez, eta nik garatzeko ditudan baldintzen aurrean liburutegiak eskaintzen dizkidanez, lengoaia hau erabiltzea erabaki dut.

Hurrengo zerrendan, lengoaia honek eskaintzen dizkugun abantaila batzuk azalduko dira:

- **Irakurgarritasuna:** Python-en sintaxia oso argia da, beraz kodea irakurtzea erraza da.
- **Objektuei orientatutako programazioa:** Objektuei orientatutako programazioarekin datu egiturak sortu ditzazkezu, horrela lana aurreztuz.

---

<sup>1</sup><https://www.python.org/>

- **Kode irekia:** Python doakoa eta kode irekikoa da. *The Python Software Foundation*-ek CPython deitutako fitxategi bitarrak doain eskaintzen ditu sistema eragile ospetsuenetarako. Iturburu kodea ere lor dezakezu, aldatu eta banatu CPython-en lizentziapean<sup>2</sup>.
- **Komunitatearen babes handia:** Python-ek komunitate aktibo handia du, web orri, posta zerrenda eta USENET "netnews" talde askorekin, lengoaiari buruzkoa ezagutza eta laguntza zabaltzeko.
- **Hedagarria:** Python-ek dituen 300 modulu estandarrez gain, badira beste doako modulu asko, lehen aipatutako komunitate handiari esker. Adibidez, edozein datu basera(MySQL, Oracle, POSTGRES,..) egokitzeko moduluak ditugu. Ondoren guk erabilitako batzuk aipatuko ditugu.

## Liburutegiak

Simulatzaila garatzeko, pythonen hainbat pakete erabili dira. Lehen esan bezala, lehenengo proiektuaren eskakizunak analizatu dira, eta ondoren, eskakizun hauei liburutegi bat bilatu zaie dena kodifikatu behar ez izateko.

Ondoren proiektuaren beharrei aurre egiteko behar izan diren liburutegiak definituko dira, gure beharren arabera:

**Simulazioa** Gure kasuan, simulazio bat egin behar izateko, denbora errealean funtziona dezakeen gertaera diskretuen simulatzaile bat behar dugu.

Hortarako, **Simpy**<sup>3</sup> liburutegia erabiltzea erabaki da. Prozesuetan oinarritutako gertaera diskretuen simulaziorako *framework* bat da. Sare asinkronoetan erabiltzeko erabil daiteke, baita ere multi-agente sistemak implementatzeko. Denbora errealean simulatzeko aukera ematen du.

**Konputazio zientifikoa** Proiektuko atal askotan, batez ere simulazioan, kalkulu zientifiko eraginkorrak egitea beharrezkoa izango da. Baita ere ausazko zenbakiak sortu beharko dira. **Numpy**<sup>4</sup> eta **Scipy**<sup>5</sup> liburutegiak behar hauei aurre egiteko funtzio guztiak ditu. Beranduago azalduko liburutegietan ere erabiltzen dira hauen barne funtzionamendurako.

<sup>2</sup>[https://en.wikipedia.org/wiki/Python\\_Software\\_Foundation\\_License](https://en.wikipedia.org/wiki/Python_Software_Foundation_License)

<sup>3</sup><https://simpy.readthedocs.io/en/latest/>

<sup>4</sup><http://www.numpy.org/>

<sup>5</sup><https://www.scipy.org/>

Lehenengoa zenbakizko konputazioarekin erlazionatua dago, bigarrena berriz, konputazio zientifikoarekin. Esan beharra dago Scipy-k numpy liburutegia erabiltzen duela bere barne kalkuluetarako.

**Ikasketa automatikoa** Edukien ospe maila kalkulatzeko ikasketa automatikoa erabili behar dugu, zehazki sailkatzaile bayesiarrak. Pythonek badu liburutegi oso sakon bat, algoritmo asko eta parametrizazio handia duena, **Scikit-learn**<sup>6</sup> izenekoak. Klasifikazio, erregresio eta *clustering* atazak egiten ditu. Pythonen zenbakizko modulu (*numpy*) eta modulu zientifikoekin (*scipy*) elkar lan egiteko diseinatua izan da.

**Bisualizazioa** Askotan emaitzak datu askokoak badira, analizatzea zaila izaten da. Análisi hori bisualki egitea errazagoa izaten da. Hortarako, **Matplotlib**<sup>7</sup> liburutegia erabiltzea erabaki da. 2D grafikoak sortzeko diseinatua izan da, eta aukera anitz eskaintzen ditu.

**Datuen Kudeaketa** Proiektu honetan zehar, garrantzitsua izango da, datuen kudeaketa eraginkorra, azkarra eta erraza egitea. Helburu hauek buruan, **Pandas**<sup>8</sup> liburutegia erabili da. Datu egiturak sortu, maneiatu eta hauen analisia egiteko erabiltzen den liburutegia da. Aurrekoak bezalaxe, aukera mordo bat eskaintzen ditu datuen kudeaketa burutzeko.

---

<sup>6</sup><http://scikit-learn.org/stable/>

<sup>7</sup><http://matplotlib.org/>

<sup>8</sup><http://pandas.pydata.org/>





## 4. KAPITULUA

---

### Proiektuaren garapena

---

Atal honetan proiektuaren garapena azalduko da, hau da, aukeratutako arazoari aurre egiteko eman diren pausoak. Horretarako, lehenik arazoa deskribatuko da, ondoren garapenaren atal ezberdinak azalduko dira. Azkenik emaitzak erakutsi dira, ohikoak diren beste sistemekin alderatuz.

#### 4.1 Arazoaren deskribapena

##### 4.1.1 Sarrera

Proiektuan ebatzi beharreko arazoa, sareetan dauden cache sistemen optimizazioan datza, ikasketa automatikoko algoritmoak erabiliz. Beste aldetik, cache horiek optimizatzeko, cache horien elkarlana bultzatuko da, emaitzak hobeak izan daitezen.

Hauek dira zehazki gure sistemak hobetu behar dituen emaitzak:

- **Cache hit ratio:** Egindako eskaera guztietatik zenbatetan asmatzen den, eta zenbatetan ez. Ehunekotan adierazi ohi da.
- **Objektuaren zerbitzatzeko denbora:** Eskaera egin denetik, zenbat denbora pasa den objektua zerbitzatu den arte.

Beste aldetik, gure sistemak honako ezaugarriak izan behar ditu:

- **Azkartasuna:** gure sistema ahalik eta azkarrena izan behar da, era horretara, caching proaktiboa ahalik eta erreaktiboena bihur dezakegu.
- **Malgutasuna:** Malgua izan behar du, hau da, parametro askotarako emaitza onak izan behar ditu, ingurune ezberdinetan ondo funtzionatzeko.
- **Automatikoki kudeagarria:** Behin sistema sortua, eta parametrizazioa aukeratuta, dinamikoki konfiguragarria izatea. Hau da, adimen puntu bat izan dezala, parametrizatzeko lana egiteko tekniko baten beharra ez izateko.

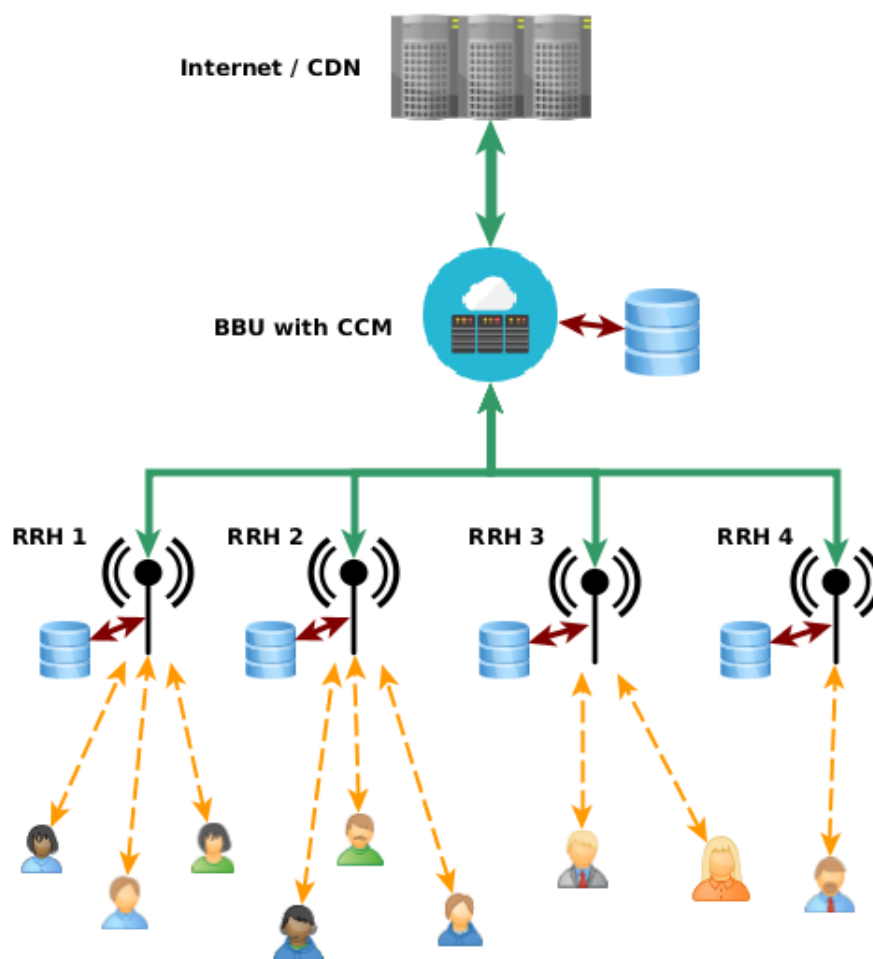
#### 4.1.2 Arkitektura

Atal honetan, cache sistema kooperatiboa probatzeko simulatu den arkitektura azalduko da.

Proposatutako sisteman, *Base Band Unit*-ean (BBU) kokatua izango da cache kudeatzailer zentrala (CKZ). Gaur egun hodeiak duen konputazio ahalmena erabiliz, erabiltzaile guztiek Radio Remote Head (RRH) izeneko antenetara eginiko eskaera guztiak monitorizatuko ditu. CKZ izango da zer cacheatzen den eta non cacheatzen den erabakitze arduraduna, monitorizazioetaz baliatuz.

Baita ere, RRH antenek cache bat izango dute, baina hauen kudeaketaz CKZ arduratuko da. Beraz, sistema hierarkiko bat edukiko dugu, non RRH antenak BBU eta hodeiarekin konektatuak egongo diren. Nahiz eta cacheatzen diren edukiak fisikoki toki ezberdinetan egon, Cache Kudeatzaile Zentralak globalki indexatutako taula bat izango du, edukien metadatuak kontrolatzeko eta cacheak kudeatzeko.

4.1 irudian agertzen da proposatutako arkitekturaren diagrama bat. Bertan, ikus daiteke erabiltzaileek RRH antenei eskaerak egiteko prozedura. Hauek cache bat dute, beraz, eskatutako edukia cacheatua balego, zuzenean edukia zerbitzatuko litzateke erabiltzaileari. RRH horren cachean ez egotean, Cache Kudeatzaile Zentralak bere taula indexatuan begiratu luke ea beste nonbaiten aurkitzen den cacheatua edukia. BBU-an bertan balego, BBU-tik RRH hortara bidaliko litzateke edukia antena horrek zerbitzatzeko. Beste RRH antena batean balego, edukia BBU-ra bidaliko litzateke hortik dagokion RRH antenara bidaltzeko. Azkenik, gure sarean katxeatua ez badago, internetera jo beharko luke edukiaren bila, horrek zerbitzatzeko dakarren luzapenarekin.



**4.1 Irudia:** *Proposatutako arkitekturaren diagrama.*

### 4.1.3 Arazoaren formulazio matematikoa

Demagun gure sareak  $\kappa = \{1, 2, \dots, k, \dots, K, BBU, CDN\}$  cache dituela konektatuak lehen aipaturiko arkitekturarekin. Beraz osotara,  $K + 2$  elementu izango dituen. Baita ere,  $\nu = \{1, 2, \dots, V\}$  eduki ditugula bakoitza bere tamainarekin,  $\zeta = \{s_1, s_2, \dots, s_V\}$  bezala errepresentatu direnak. Azkenik eskaeren multzo bat ere dugu,  $\rho = \{1, 2, 3, \dots, P\}$  bezala definituta.

Hau jakinda, honako aldagaiak definitu ditugu, jakinda  $\nu \in \nu$  eta  $f, t \in \kappa$  direla:

$$P_{\nu ft} = \begin{cases} 1, & \text{baldin } \nu \text{ edukiaren eskaera dagoen } f\text{-tik } t\text{-ra} \\ 0, & \text{bestela} \end{cases} \quad (4.1)$$

$$T_{ft} = \begin{cases} 0, & \text{baldin } t = f \\ \varepsilon > 0, & \text{baldin } t = BBU \\ E > \varepsilon, & \text{baldin } t \neq \{f, BBU, CDN\} \\ \xi > E, & \text{baldin } t = CDN \end{cases} \quad (4.2)$$

$$C_{\nu t} = \begin{cases} 1, & \text{baldin } \nu \in t \\ 0, & \text{bestela} \end{cases} \quad (4.3)$$

4.1 ekuazioan, eskaerak errepresentatzen dituen aldagaia definitu da. 4.2 ekuazioan berriaz, edukia zerbitzatzeko denbora definitzen da. Azkenik, 4.3 ekuazioak edukia bat non cacheatua dagoen zehazten du.

Pasatako denbora leihore batean egindako eskaerak kontuan hartuz, zerbitzatzeko denbora minimizatzeko, honako formulazioa definitu da:

$$\min t = \sum_{p=1}^{\rho} \sum_{f=1}^K \sum_{t=1}^{K+2} p_{\nu ft} T_{ft} C_{\nu t} \quad (4.4)$$

hauen mende:

$$\sum_{t=1}^{K+2} C_{\nu t} = 1, \quad \forall \nu \quad (4.5)$$

$$\sum_{v=1}^v S_v C_{vt} \leq S_t, \quad \forall t \quad (4.6)$$

Lehenengo menpeko ekuazioak, eduki bakoitza bakarrik leku batean cacheatua izan daitekeela dio. Guk jarritako murriztapen pertsonala da, etorkizunean posiblea izango litzateke hori aldatzea, eta algoritmo mamitsuagoak erabiltzea behin murriztapen hau kendurik.

Bigarrenak berriz, cache bakoitzaren tamaina errespetatzearen murriztapena zehazten du. Hau da, cache berdinean dauden eduki guztien tamaina ezin da cachearen tamaina baino handiago izan.

## 4.2 Metodoa

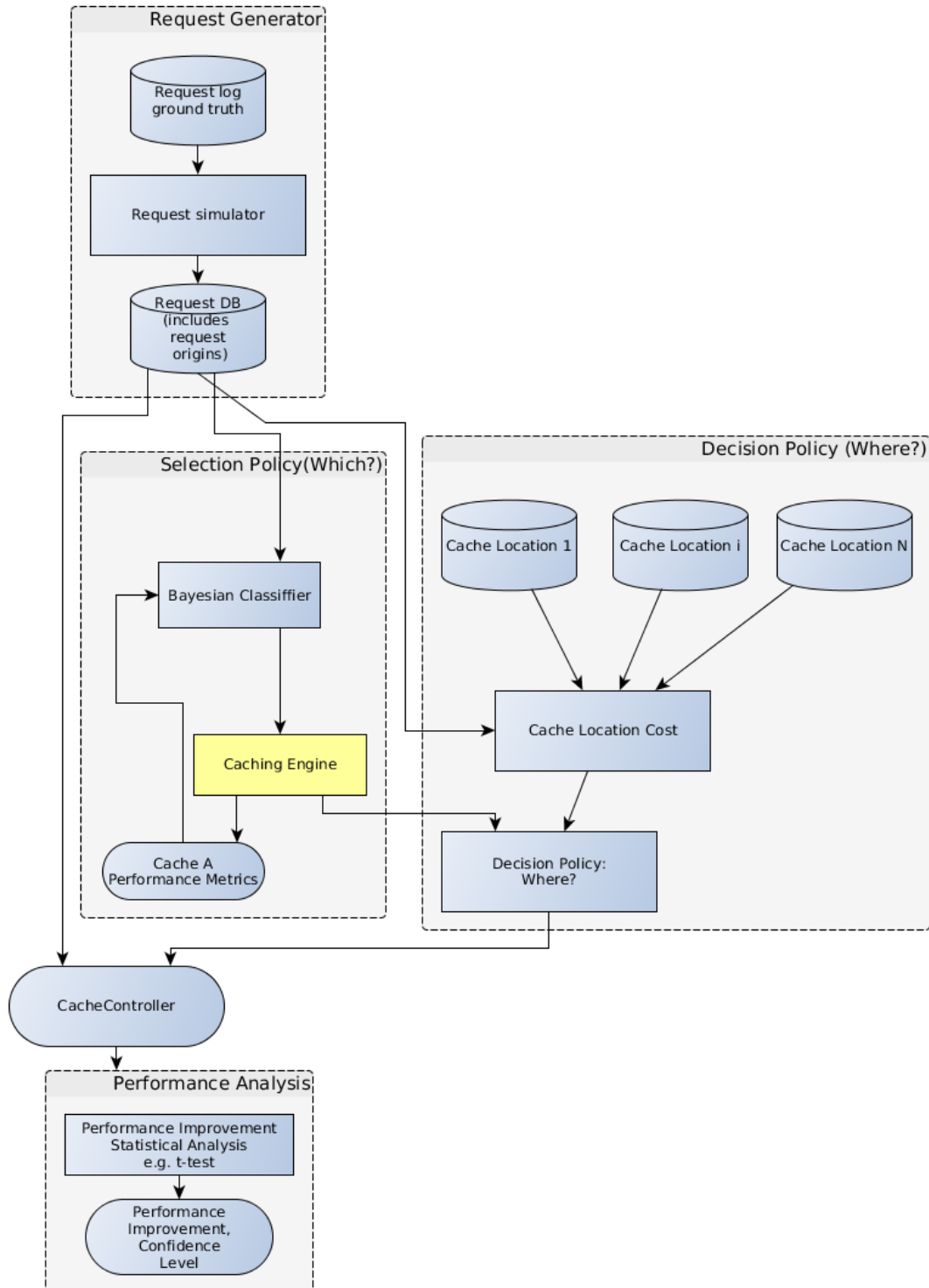
Arazoa ebazteko, metodo bat proposatu dugu, algoritmo jale batez baliatuz, honen barruan sailkatzaile bayesiarrak erabiliz. 4.2 irudian ikus daiteke erabilitako sistema edo metodoaren mapa kontzeptuala.

Algoritmo jaleek hiru elementu komun izaten dituzte:

- **Hautapen prozesua:** oraindik aukeratu ez diren hautagaietatik soluzio onenaren bideratzailea den hautagaia aukeratuko duen prozedura.
- **Helburu funtzioa:** soluzio bati dagokion balioa edo kostua itzultzen duena.
- **Osotasun funtzioa:** aukeratutako azken hautagaia tarteko emaitzari erantsigarria den ala ez erabakitzen duena.

Lehenik cache sistamarako bi modulu nagusi bereiz ditzakegu, lehena ospe mailaz edukiak sailkatuko dituena. Bigarrena, ospe maila honen arabera edukien kokapen optimoa egiten saiatuko da. Ondorengo ataletan hauek hobeto espezifikatuko dira.

Aurretik azaldutako algoritmo jaleen bi elementu identifikatu daitezke gure metodoan. Lehenik, ospe mailaren arabera edukiak sailkatuko dira. sailkapen hori erabiliko da ondoren edukiak aukeratzeko tarteko soluzioa lortzeko. Osotasun funtzioa, kokapen sistemaren barnean erabiliko da, hau da, eduki bat kokatzean, soluzio hori zuzena ez bada, hurrengo tarteko soluzio optimoa bilatuko da, eta horrela tarteko soluzioa zuzena izan arte.



4.2 Irudia: Erabilitako metodoaren mapa kontzeptuala

### 4.2.1 Hautapen prozesua: Edukiak ospe mailaren arabera sailkatu

Atal honetan, algoritmo jalearen zati mamitsuena dago. Hemen edukiak ospe mailaren arabera sailkatuko dira. Ondoren kokapenaren optimizaziorako hautapen prozedura bezala ospe mailarekin kalkulaturako funtzio bat erabiliko da.

Horretarako, jasotako eskaeren taula bat izango dugu Cache kudeatzaile zentralean. Ondorengo datuak bilduko ditu:

- **Pr:** Edukiak jasotako eskaera guztiek, etorkizuneko denbora leiho batean beste eskaera bat jasotzeko probabilitateen batura. Hau da, eskaera bakoitzaren datuen arabera, probabilitate bat kalkulaturako du sailkatzaile bayesiarrak. Denboran zehar eduki horrek jaso dituen eskaera bakoitzaren probabilitateak batuko dira. Float motakoa.
- **Result:** 4.7 ekuazioan azaltzen den formula erabiliz kalkulaturako da. Hau izango da ondoren edukiak ordenatzen erabiliko den balioa. Float motakoa
- **Freq:** Edukiak jasotako eskaera kopurua. Integer motakoa.
- **Recency:** Eduki horrek jasotako azken eskaeraren ordua. Gure simulatzailearentzat float modukoa, denbora segundutan kalkulatzeko duelako, baina posiblea da baita ere beste denbora datu motak erabiltzea.
- **Size:** Edukiak duen tamaina. Float motakoa.

Datu hauek *Pandas*-en DataFrame baten barruan joango dira, eta edukiaren kodearekin joango dira indexatuak. Horrela, edukiak iristean, hauen bilaketa azkarragoa izango da DataFrame-aren barruan.

Eskaera bat iristean, Cache Kudeatzaile Zentralak eskera hau gordeko du, eta eduki horren dautak eguneratu. Baita ere, eskaera bakoitzaren denbora leihoa pasa denean eskaera hori sailkatzailean entrenatuko da, eduki horrek ondoren eskaera bat jaso duen ala ez definituz. Sailkatzaile osoa ez entrenatzeko, guk erabilitako *Machine Learning* liburutegiak funtzio bat eskaintzen digu partzialki entrenatzeko, hau da, ez dira eskaera guztiak entrenatuko, eskaerak aurretik entrenatutakoari gehitzea eskaintzen digu. Modu hontara, gure sailkatzailea ia denbora guztian guztiz entrenatua egongo da.

Result zelaia definitzerako orduan aipatu den adin faktorea (*af*), Cache Kudeatzaile Zentralak gordetzen duen aldagai global bat izango da. Adin faktorea kalkulatzeko, cachetik kanpo geratu den lehen edukiaren Result zelaiaren balioa izango da.

Result zelaia era hontara kalkulatu da:

$$Result = af + \frac{Pr}{Size} \quad (4.7)$$

Sailkatzaile bayesiarra

Sailkatzaile bayesiarrak oso erabiliak dira gaur egun datu meatzaritzan, sailkatzaile probabilitistikoak dira, Bayes-en teoreman oinarritutakoa. Gure sailkatzailea Naïve Bayes du oinarri bezala.

Naïve Bayes atributuen arteko erlazioak independenteak direla du hipotesi bezala, beraz konputazionalki oso azkarra izango da. Gure kasuan, Naïve Bayes-en algoritmoaren zati bat erabiliko dugu, gure beharra objektuak ordenatzea denez, hortarako, beste eskaera bat jasotzeko probabilitateak lortuko ditugu.

4.1 taulan azaltzen dira erabilitako atributuak. Klase modura, etorkizuneko denbora leiho baten objektu horren eskaera bat egon den ( $c = 1$ ) edo ez ( $c = 0$ ) ezarri dugu.

Atributua	Deskribapena	Mota
$X_1$	Pasatako denbora azken eskaeratik	Jarraitua
$X_2$	Eskaeren maiztasun denbora leiho batean	Osokoa
$X_3$	Eskaeren maiztasuna denbora leiho luzatu batean	Osokoa
$X_4$	Objektuaren tamaina	Jarraitua

**4.1 Taula:** Sailkatzaile bayesiarraren atributuak

Beraz, gure kasuan,  $c = 1$  interesatzen zaigu, beraz bakarrik probabilitate hori kalkulatu dugu.

### Gaussian naive bayes

Datu jarraituekin lan egitean, ohikoa da klasearekin erlazionatutako atributu jarraituak banaketa gausiarra dutela suposatzea. Hortik abiatuz, dataseta klasearen arabera segmentatu, batzbestekoa eta bariantza kalkulatu da klase bakoitzerako.

Izan bitez  $\mu_c$  eta  $\sigma_c^2$   $x_i$  atributuak duen batzbestekoa eta bariantza  $c$  klasearekiko. Beraz,  $v$ -ren probabilitate banaketa  $c$  klasearekiko honako eran formulatu daiteke:

$$p(x = v|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(v-\mu_c)^2}{2\sigma_c^2}}$$



Behin atributu batentzat probabilitatea jakinda, Naïve Bayes-en teknika berbera aplikatu dugu, non atributuen arteko erlazioak kontuan hartzen ez ditugunez, hauen probabilitateak biderkatu beharko ditugu soilik:

$$p(X_1 = x_1, \dots, X_n = x_n | c = 1) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(x_i - \mu_c)^2}{2\sigma_c^2}}$$

#### 4.2.2 Helburu funtzioa: Kokalekua Bilatu

Behin edukiak sailkatuta, algoritmo jaleak edukiak banan banan hartuko ditu eta kokatu egingo ditu. Atal hontan kokatzeko prozesu hori azalduko da.

Lehenengo, eduki horrek izan dituen eskaerak jasoko dira, beti ere aurretik definituriko denbora leiho bakarrean. Beraz, denbora leiho horretan eginiko eskaeren arabera, edukiaren kokalekua erabakiko da. Hau da, eskaera hauek zerbitzatzeko denbora hoberena duen cachean kokatuko da. Cache hori betea badago, zerbitzatzeko denbora gutxien duen hurrengo cachean cacheatzen saiatuko da. Horrela, edukiak cacheatzen joango da cacheak bete arte.

2 algoritmoan ikus daiteke funtzio honen sasikodea.

#### 4.2.3 Egitura

Klase Diagrama

4.3 irudian ikus daitekeenez, *NBCachingEngine* klasea da klase nagusia. Honek *CRAN* objektu bat gordetzen du, kokapen optimoa lortzeko arkitekturaren datuak behar dituelako. Horretaz gain, *Classifier* klasea dugu, non sailkatzaile bayesiarra dugun. *video\_dataset\_generator* klaseak edukien tamainak modelatzen ditu. Bakarrik simulaziorako erabiliko da. Etorkizunean modulu hau ezabatu edo bideoen tamaina lortzeko moduluren batekin ordezkatu beharko litzateke.

### 4.3 Proba Ingurunea: Simulatzailea

Probak egiteko, ezin izan dugunez lortu benetako ingurune batean algoritmoa kokatzea, eta ez dugunez bilatu gure beharrentzako egindako *testbed*-ik, gure simulatzaile propio

**Data:** eskaerak: Ilara motakoa;

cran: CRAN objektua

condition: Harien arteko komunikaziorako FLAG-a

eduki\_taula: CKZ-ko edukien DataFramea

HASIERA

eduki\_taula ordenatu Result-en arabera

**while** *eduki guztiak ez tratatuak* **do**

    hartu hurrengo edukia

**while** *Cache guztiak ez azterturak* **do**

        | Kalkulatu zerbitzatzeko denbora cache hortan

        | Hurrengo cachea

**end**

    Ordenatu Cachea zerbitzatzeko denboren arabera

    Hartu zerbitzatzeko denbora minimoa duen cachea

**while**  $tam(edukia) + tam(Cache) > Cache$  *edukiera* **do**

        | Hurrengo cachea

**end**

**if** *Badago tokia cacheren batean* **then**

        | Kokatu edukia cache hortan.

**else**

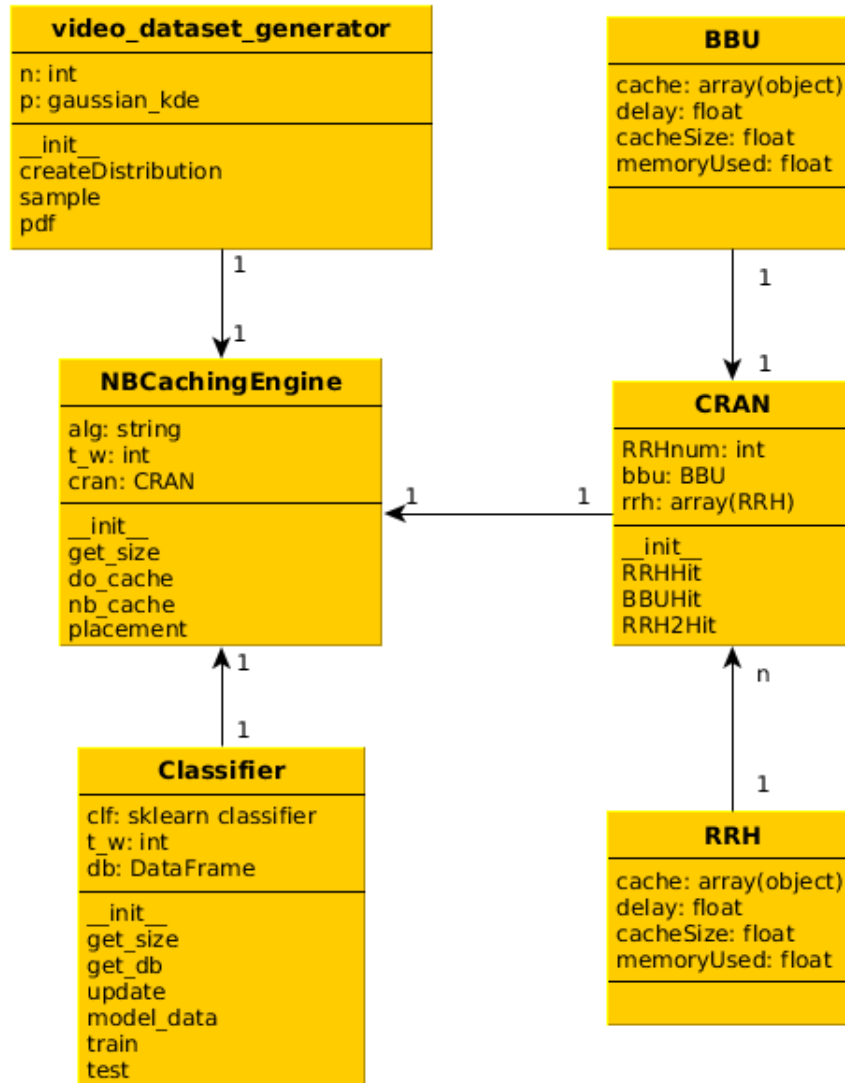
        | Itzuli cacheatu den edukia adin faktorea kalkulatzeko

**end**

**end**

Itzuli 0 adin faktorea kalkulatzeko

**Algorithm 2:** Kokalekua bilatzeko algoritmoa



4.3 Irudia: Caching sistemaren klase diagrama

bat egin behar izan dugu.

Simulatzailerik honek ondoko ezaugarriak izan behar ditu:

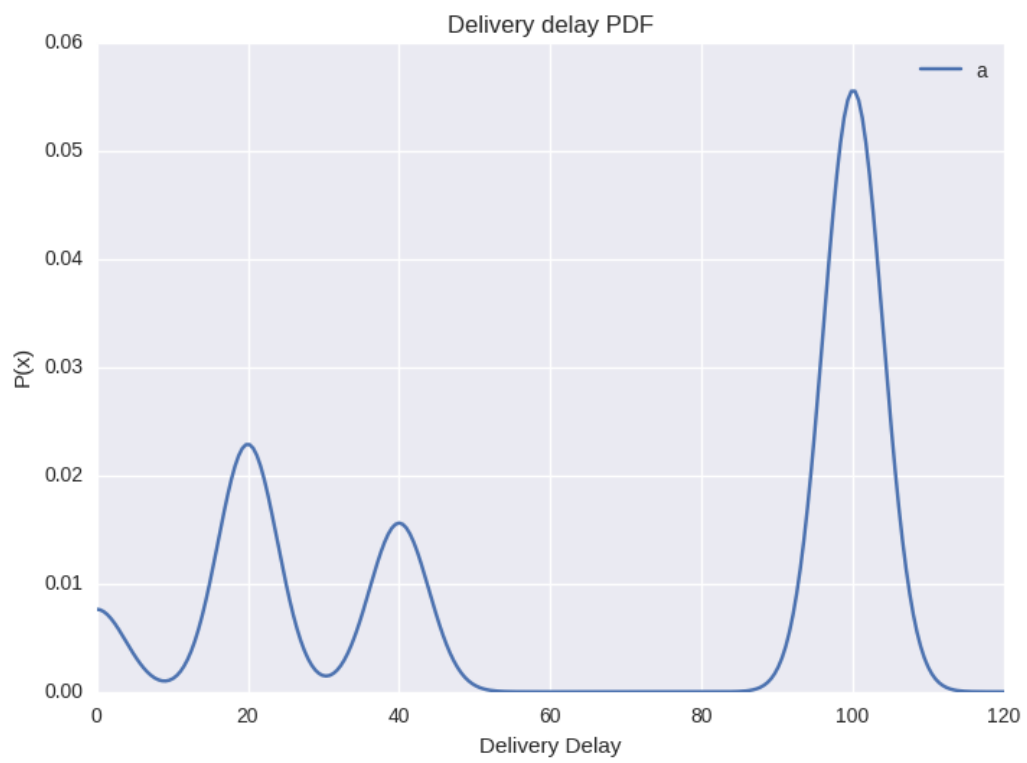
- Aurretik aipatutako arkitektura islatzea.
- Malgutasuna eskaintzea (Hainbat parametroekin jolastu ahal izatea).
- Eskaera sortzaile bat edukitzea, errealitatean oinarritua.
- Objektuen pisuak, errealitatean izaten dituzten pisu banaketaren arabera banatzea.
- Denbora errealean probak egin ahal izatea, algoritmoen azkartasunaren eragina ere atzemateko.

Ezaugarri hauek edukitzeko, sarrerako parametro hauek definitu dira:

- RRH antena kopurua.
- Nodoen arteko latentzia. (Zenbat denbora igarotzen den nodo batetik bestera informazioa eramaten).
- BBU-ko eta RRH-ko cacheen tamaina.
- Ospe mailaren arabera sailkatzeko algoritmoa.
- Objektuak kokatzeko algoritmoa.
- Eskaerak sortzeko behar diren parametroak.

Gure algoritmoen eragina behatzeko, hurrengo irteera parametroak definitu dira:

- *Cache hit ratio* batazbestekoa.
- *Delivery Delay* batazbestekoa.
- *Delivery Delay*-ren probabilitate dentsitate funtzioaren grafiko bat. 4.4 irudian ikus daiteke adibide bat.



**4.4 Irudia:** Irteerako Delivery Delay-ren Probabilitate dentsitate funtzioaren adibide bat.

### 4.3.1 Harien arteko komunikazioa

Beste arazo bat izan dugu: Nola komunikatu gure simulatzailea gure algoritmoarekin. Denbora errealean exekutatzean, bi hari ezberdinetan exekutatu behar izan da, denboraren ikuspegitik batak bestearen exekuzioan eragina ez izateko.

Python-ek hortarako modulu estandar bat eskaintzen du **threading** izenekoa. Bi hariiek datuak partekatzen dituztenez, modulu honek duen *condition* izeneko objektu bat erabili da blokeoak egiteko algoritmo baten eta bestearen artean. Bi funtzio erabili dira nagusiki, *wait* eta *notify*.

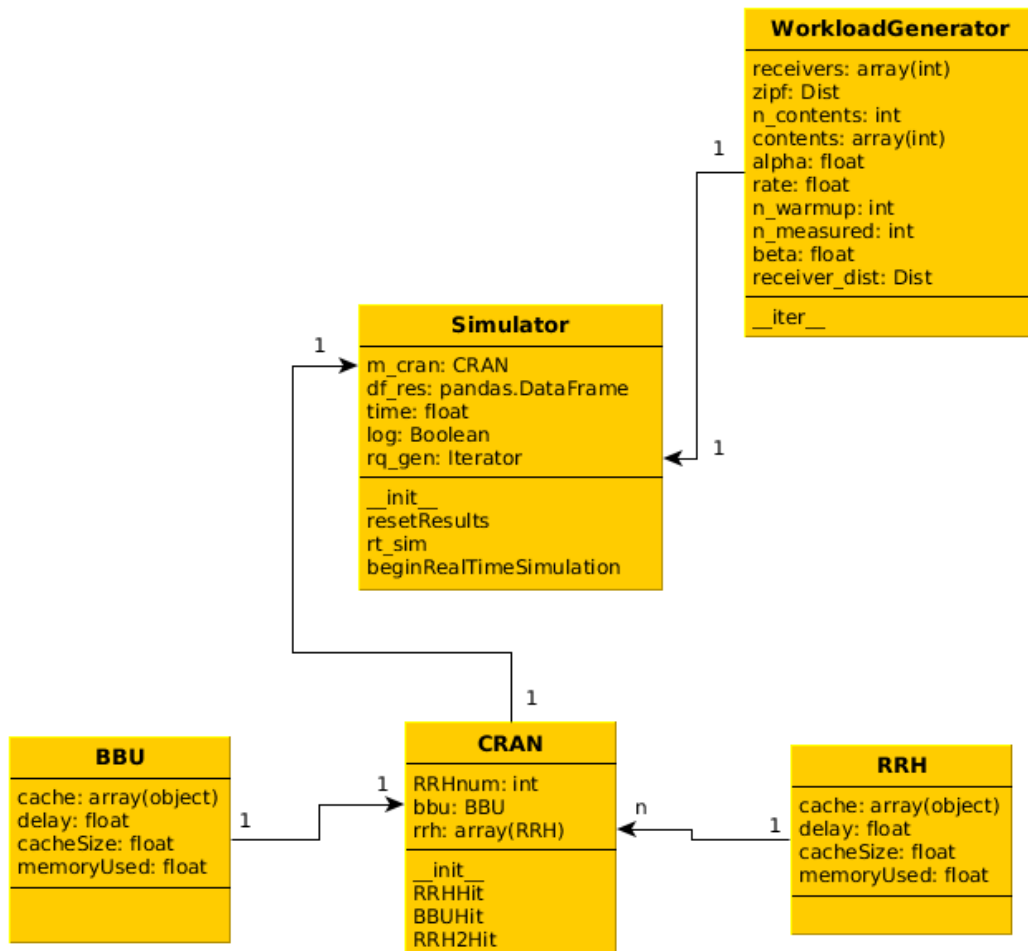
Lehenengoa beste hariiek blokeoa askatzeko zain geratzeko balio du, beraz beste hari batek blokeoa eskatzen duenean, honek blokeoa hartuko du. Bigarrenak, *notify* funtzioak, blokeoa eskatuko du eta exekuzioan aurrera jarraituko du. Bi funtzio hauek erabili dira harien komunikazio kostua jaisteko.

Hauek izan dira partekatzen dituzten aldagaiak:

- **cran objektua:** Simulatzaileak hau erabiltzen du zein eduki eta zein cachetan cacheatua duen jakiteko. *CachingEngine*-k hemen idatziko ditu datu horiek.
- **r\_queue:** Eskaera ilara bat da. Simulatzaileak sortzen dituen eskaerak gure algoritmoari pasa behar dizkio, eta honek eskaerak erabili behar ditu jakiteko zein eduki eta non cacheatu behar diren erabakitzeko.

Hauek dira *wait* eta *notify* funtzioak erabiltzen diren kasuak:

1. **Simulatzailean:** Eskaera berri bat sortu baino lehen, simulatzailea blokeoa askatzeko zain geratuko da, eskaera ilaran eta *cran* objektuan konfliktua ekiditeko. Behin cachean edukia dagoen egiaztatu ondoren eta eskaera ilaran sartuta, *notify* funtzioa erabiliko da blokeoa askatzeko, beste funtzio batzuk egiten dituen bitartean.
2. **CachingEngine:** *Wait* funtzioari dei egingo zaio eskaera ilara hustu baino lehenago. Bere txanda denean, eskaera guztiak gordeko ditu eta *notify* funtzioa erabiliko da simulatzaileari blokeoa pasatzeko. Bitartean cachearen kalkulu guztiak egingo ditu eta behin hauek eginda, simulatzaileak blokeoa askatzeko zain geratuko da, *cran* objektuan cachearen egoera idatz dezan. Behin idatzita, *notify* funtzioa erabiliko da blokeoa askatzeko eta aurrera jarraitzeko.



4.5 Irudia: Simulatzailearen klase diagrama

### 4.3.2 Egitura

#### Klase Diagrama

4.5 irudian azaltzen da simulatzailearen klase diagrama.

Ikus daitekeenez, klase nagusia *Simulator* da, eta honek bere barnean *WorkloadGenerator* objektu bat du. Objektu honek eskaerak sortzen dizkigu.

Baita ere, *CRAN* klaseko objektu bat gordetzen du, nork aldi berean *BBU* objektu bat eta hainbat *RRH* objektu dituen. *CRAN* klasea, arkitektura modelatzeaz arduratzen da, eta logikoki bere barnean bai *RRH*-ak eta *BBU* bat izan behar ditu.

## Simulator klasea

4.5 irudiko klase diagraman ikus daitekeenez, klase nagusia *Simulator* klasea da. Hor barnean, Simpy paketeko simulazioa sortzen da, eta *rt\_sim* funtzioa da exekutatzen dena simulatzaile bezala. 3 algoritmoan azaltzen da azken funtzio honen pseudokodea.

```
Data: r_queue: Ilara motakoa;
cran: CRAN objektua
condition: Harien arteko komunikaziorako FLAG-a
log: boolearra
while simulazio denbora ez da amaitu do
    Eskaera sortu;
    Itxoin eskaeraren denbora iritsi arte;
    Blokeoaren zain geratu;
    Begiratu non egin den Hit bat;
    Eskaera ilaran sartu;
    Blokeoa askatu;
    Log-ean gorde;
end
```

**Algorithm 3:** Simulatzailearen algoritmo nagusia

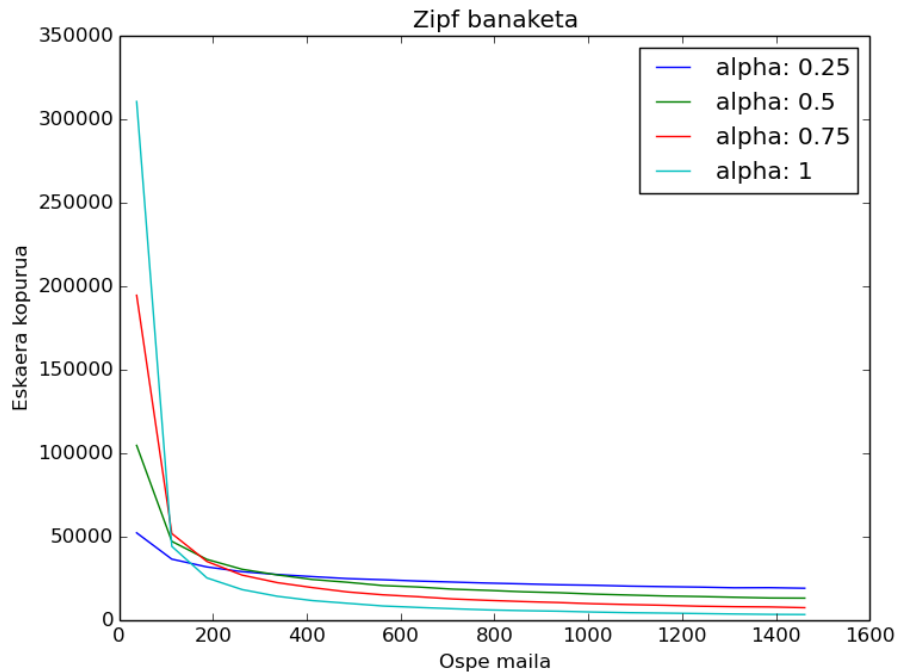
## CRAN klasea

Lehen aipatu bezala, klase hau arkitektura modelatzen duena da. Klasea sortzean sartzen diren parametroekin bakarrik konfiguratzeko da. Hau da, RRH antena kopurua, cacheen tamaina eta nodoen arteko latentzia gordetzen ditu. Edukia non cacheatuta dagoen eta zerbitzatu arteko denbora kalkulatu dituzten funtzioak *RRHHit*, *BBUHit* eta *RRH2Hit* dira. Klase hau ere CachingEngine moduluak erabiliko du, batez ere objektuen kokapenetan erabakitzeko eta nodo bakoitzaren cacheak berrantolatzeke, hemen idatziko baititu cachean sartu diren edukiak.

## Workload Generator klasea

Simulatzailea aurrera eramateko, eskaera sortzaile bat behar genuen, eta ahal zen moduan, sistema errealista bat simulatzen zuena. [BCF<sup>+</sup>99] artikuluak definitzen du ospean oinarritutako eskaerek interneten izaten duten banaketa. Lan honen arabera, normalean eskaerek Zipf-en Banaketa jarraitzen dute. Banaketa honek bi parametro ditu, beraz gure beharren arabera parametro horiek aldatzen joan gaitzke.





**4.6 Irudia:** Zipf banaketa  $\alpha$  ezberdinekin

Lehen parametroa  $\alpha$  da, ospe banaketa zehazten duen parametroa da. Alpha balioa handia denean, eskaera gehienak ospe maila gehien duten edukietan kontzentratuko dira, eta alpha txikia denean, eskaera nahiko banatua izango da eduki guztien artean. 4.6 irudian adibide bat ikus daiteke, alpha ezberdinetarako ospe mailaren arabera izandako eskaera kopuruekin.

Bigarrena  $n\_contents$  atributua da. Azken honek, eduki kopurua definituko du.  $Beta$  atributuak RRH antenen arteko eskaeren banaketa modelatzen du, beste Zipf banaketa bat erabiliko du, antenak ere ospe maila bat balute bezala.

Beste atributu nagusien artean,  $rate$  aurkitzen da, segunduko zenbat eskaera sortu behar diren adieraziko duena. Hau ez da zehatza, hau da, banaketa esponentzial baten aldaera bat jarraitzen du.

## Edukien tamainaren simulazioa

Edukien tamaina modelatzeko, interneten dataset bat lortu dugu, Congas project<sup>1</sup> izeneko proiektuan eskaintzen dena. Beraiek sortutako YouStatAnalyzer erraminta erabiliz lortua. JSON fitxategi batean datoz bideo hauen datu guztiak, beraz, bideo hauek parseatzeko aplikazio txiki bat sortu behar izan da. Hortik, bideoen iraupena dataset batean gorde da.

Dataset hori, gure edukien artean banatuko da, Nukleoaren dentsitate estimazio algoritmo batekin sortuta. Algoritmo hau, probabilitate dentsitate funtzioak (PDF) estimatzeko erabiltzen da. Beraz, lehenengo datu hauen PDF-a sortzen da, eta probabilitateen arabera, edukiei iraupen bat esleitzen zaie. Beraz, edukiak, dataset honek duen iraupenen probabilitate dentsitate antzekoa izan behar du.

Bideoetan oinarritu garenez, iraupena erabiliko dugu tamaina kalkulatzeko. Dataset honetan ere *bitrate*-a azaltzen da, eta ia bideo gehienek 320kbps-koa zutenez, bideo guztiei bitrate hori esleitu zaie kalkulua egiteko. Tamaina, iraupena eta bitratea bidertzen lortuko dugu.

## 4.4 Emaitzak

Hainbat simulazio burutu dira gure metodoaren eraginkortasuna analizatzeko. Aurreko ataletan hainbeste aipatu den parametrizazioa kontuan hartu da. Hiru parametro hauen arabera lortu ditugu emaitzak:

1. **Ospe banaketaren arabera:** Zipf distribuzioa defintzeko beharrezkoa den alpha parametroaren arabera simulazioak gauzatu dira, ospearen eragina aztertzeko caching algoritmo ezberdinekin. Gero eta alpha handiagoa, edukien ospea pikoagoa izango da, hau da, ospe gehien duten eduki gutxi batzuk eskaera gehienak jasoko dituzte, eta besteek gutxiago.
2. **Cachearen tamainaren arabera:** Cachearen tamaina garrantzitsua da beti ere, honen arabera eduki gehiago edo gutxiago cacheatu daitezkeelako.
3. **Denbora leihoaren arabera:** Gure algoritmoak denbora leihoa erabiltzen duenez, honen egokitasuna aztertzea ere garrantzitsua izan daiteke.

---

<sup>1</sup><http://www.congas-project.eu/>

Parametroa	Balioa
Simulazio Denbora	12000 segundu
RRH antena kopurua	3
RRH ->BBU Zerbitzatzeko denbora	20ms
BBU ->Internet Zerbitzatzeko denbora	80 ms
Bideo Kopurua	10000
RRH antenatarako eskaera banaketa	Uniformea

**4.2 Taula:** Simulazioen parametro konstanteak

Gure metodoa beste bi ohiko metodorekin konparatu da, *Least Recently Used* eta *Least Frequently Used*.

Gainera, grafiko bakoitza hiru azpigrafikotan banatuta dago, simulatzaileak dituen irteera parametroak izanik azpigrafiko bakoitzean azalduko dena. Hauek dira hurrenez hurren, azpigrafiko bakoitzean azaltzen diren emaitzak:

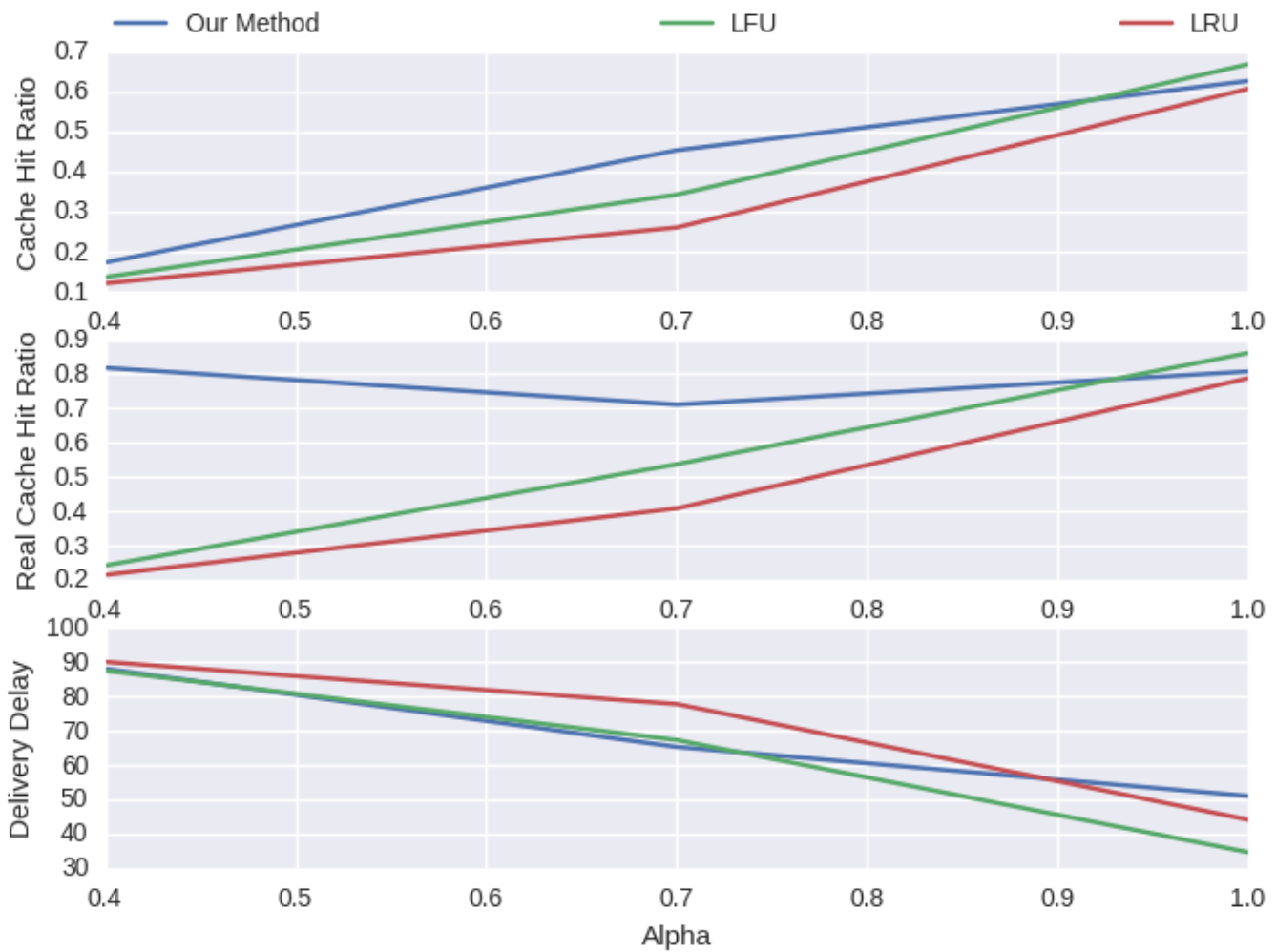
- **Cache hit ratio:** Zenbat den cacheetan izan den asmatze tasa.
- **Real Cache hit ratio:** Aurrekoaren antzekoa, baina eduki bakoitzaren lehendabiziko eskaerak kendu dira, hauek ezinezkoak direlako cacheatzea informazio gehiago erabili gabe.
- **Delivery Delay:** Batazbestez, zenbat denbora behar izan duen edukiak zerbitzatzeko.

Simulazioen parametro konstanteak [4.2](#) taulan azaltzen direnak izango dira.

#### 4.4.1 Ospe Banaketaren arabera

Lehen azalduko Zipf banaketak eraginak izan ditzake gure algoritmoaren funtzionamenduan, beraz 3 alpha desberdinetarako (0.4, 0.7 eta 1) exekutatu ditugu algoritmoak. Alpha gero eta txikiagoa izan, edukien eskaerak banatuagoak izango dira eta gero eta altuagoak izan, gero eta eskaera gehiago konzentratuko dira ospe gehien duten edukietan. Beste parametro guztiak konstante utzi ditugu, benetako ondorioak atera ahal izateko.

[4.7](#) irudian azaltzen dira emaitzak.



**4.7 Irudia:** *Emaitzak alpha ezberdinetarako.*

Emaitzak hauetatik ondoriozta daitezke alpha handia denean ez, gure metodoak asmatze tasa onak lortzen dituela, baina zerbitzatzeko denbora orokorrean ez dela oso ona.

Lehenengo azpigrafikoan, Cache hit Ratioari dagokiona, ikus daitezke, gure metodoak emaitza hobekak izaten dituela alpha txikiak direnean. Ospe banaketa pikoa denean, emaitza okerragoak lortzen ditu.

Hurrengo azpigrafikoan cache hit ratio eta cache hit ratio errearen ezberdintasuna ikus daitezke. Edukiak oso banatuak daudenean, eduki gehiago eskatzen dira, eta hor gure algoritmoak, bigarren eskaeratik aurrera, emaitza oso onak ditu. Ia eskaeren %80tan asmatzen du. Edukien ospearen banaketa erdi mailan dagoenean, emaitzak ere hobetzen direla iragarri daitezke, LFU algoritmoa baino %20 hobea delako. Aurreko azpigrafikoan bezala, ospe banaketa pikoa denean emaitza LFU algoritmoa baino okerrago dabil. Hemen ere argi ikus daitezke nola bi azpigrafikoetan, hiru algoritmoen arteko distantzia oso antzekoa den. Alpha hain handia denean, eskatutako eduki kopurua baxua delako, eta Cache hit ratio eta Real cache hit ratioen arteko ezberdintasuna oso baxua delako, lehenengo aldiz egiten den eskaera kopurua ere askoz baxuagoa delako.

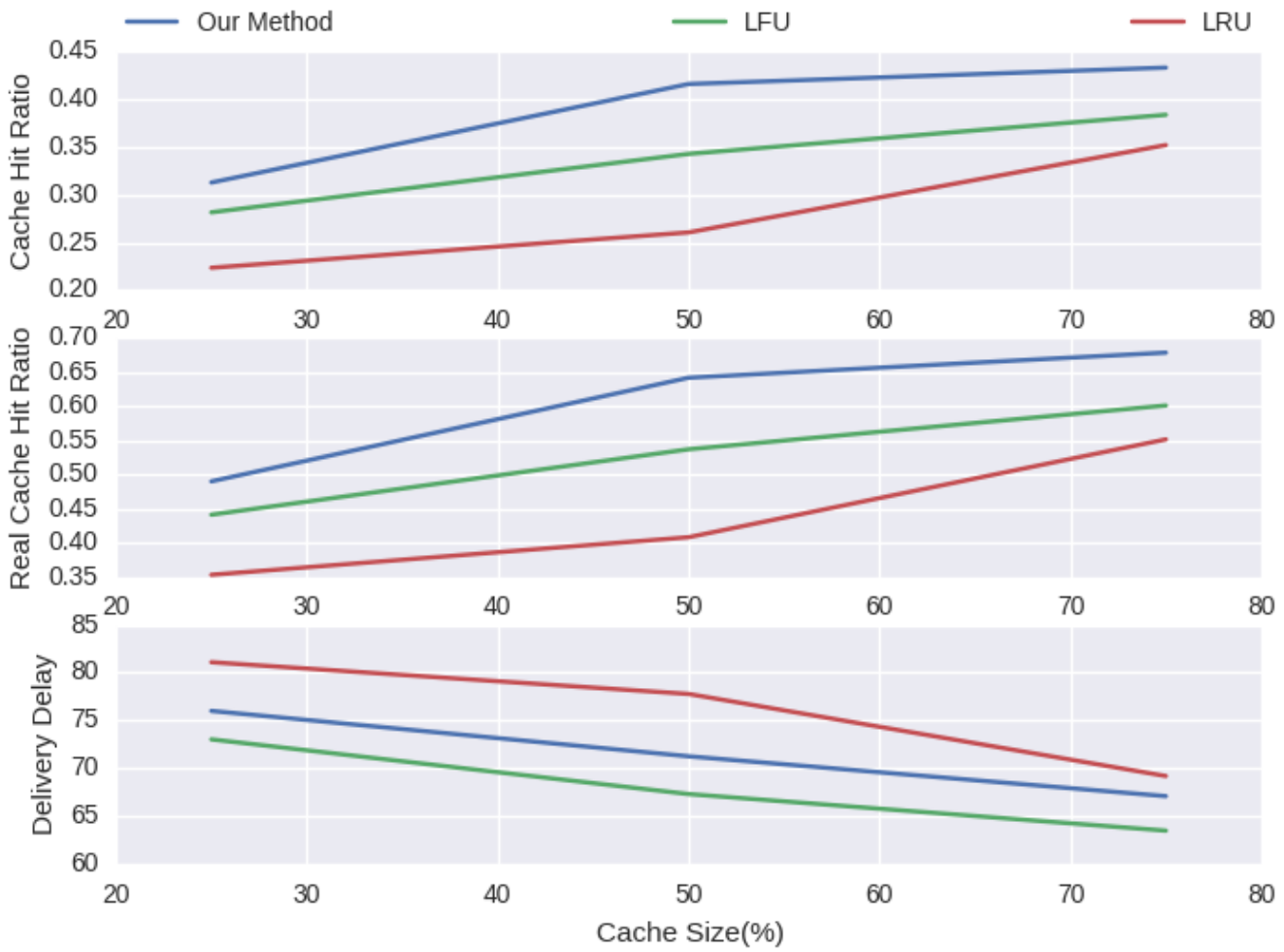
Zerbitzatzeko behar izan duen denbora da azken azpigrafikoan irudikatzen dena. Orokorrean, gure algoritmoaren emaitzak okerragoak dira. Alpha baxua dugunean, Cache Hit ratio handiago da, baina Delivery Delay-a LFU algoritmoarekin parekatuta dago, LRU emaitza zerbait okerragoak lortuz. Alpha balio ertain bat dugunean, balio txikietan gertatzen den bezala, zerbitzatzeko denbora antzekoak dituzte, nahiz eta kasu hontan LFU algoritmoa oso gutxian hobetu. Hala ere, Cache Hit Ratioek duten diferentziarekin, zerbitzatzeko denbora hain berdina izateak, gure algoritmoak lortutako Delivery Delay emaitza txarra iragartzen du. Azkenik, alpha altuerako, beste grafikoetan ez bazituen emaitza oso onak lortu, asmatze tasen eta zerbitzatzeko denboraren erlazioa jarraituz, alor honetan ere emaitza okerragoak lortu ditugu.

#### 4.4.2 Cache Tamainen arabera

Cachearen tamaina begiratzuz, ondorio asko atera ditzazkegu. Hiru tamaina ezberdinekin probatu dira algoritmoak, gutxi gora behera, cache gutzieren artean edukien %25, %50 eta %75a errepresentatzen duten tamainekin. BBU-ak duen cachea logikoki handia izango da, kasu hontan RRH antenen cacheen bikoitza izango litzateke.

4.8 irudiko grafikoan azaltzen dira emaitza hauek.

Lehen azpigrafikoa ikusita, Cache Hit ratioari dagokiona, gure metodoak emaitza onak



**4.8 Irudia:** *Emaitzak cache tamaina ezberdinetarako.*

lortzen dituela esan genezake. Aurreko emaitzen bide berbera jarraitzen du, asmatze tasa altuagoak ditu, baina zerbitzatzeko denbora zerbait okerragoa lortzen du.

Asmatze tasei dagokienez, logikoa denez, gero eta cache handiagoa izan, gero eta asmatze tasa altuagoak lortu dira. Ikus daitekeenez, cache tamaina handitzen den bitartean, algoritmoen asmatze tasen artean diferentzia pixka bat handitzen direla. Gure metodoaren emaitzak linealki hazten dira, hau da, erlazio lineala du cache tamainak gure metodoaren asmatze tasekin.

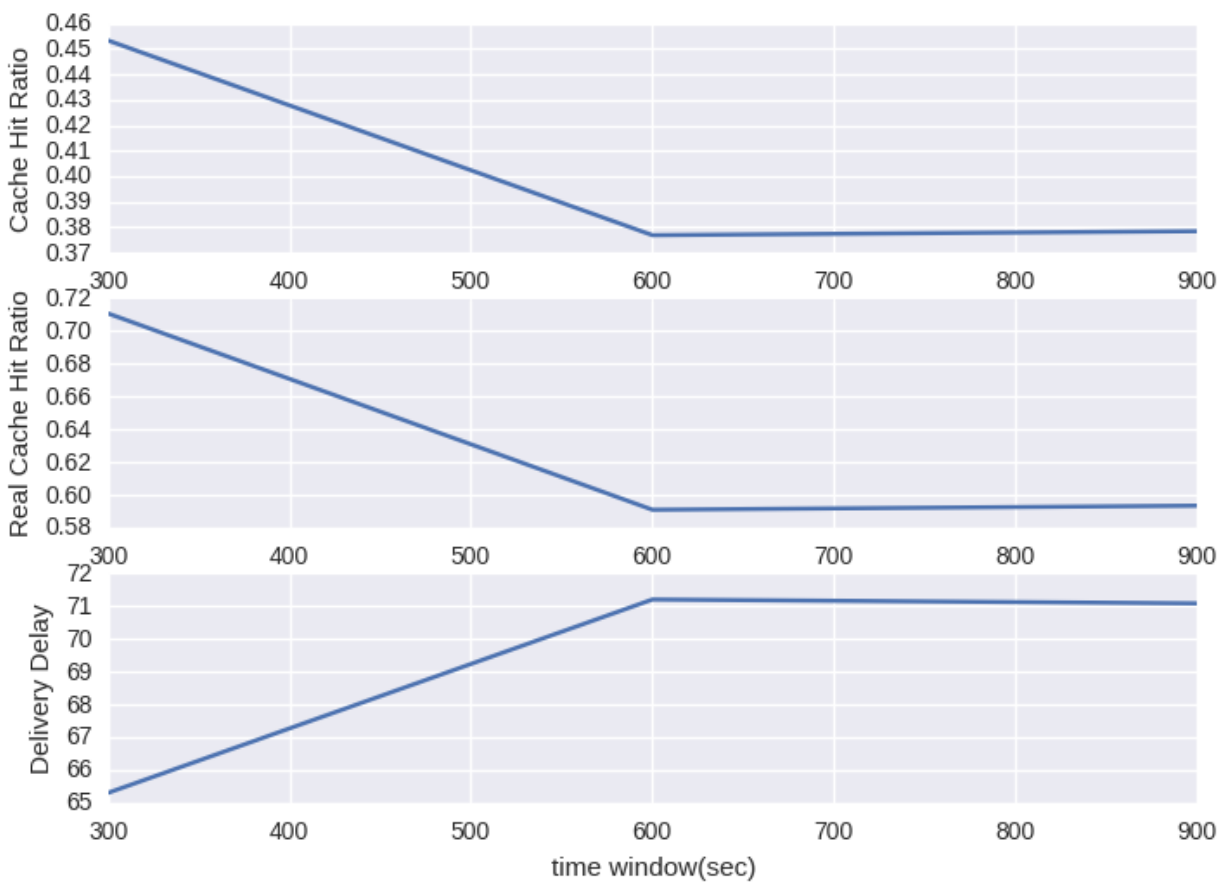
Edukiak zerbitzatzeko denborari dagokionez, aurretik aipatu bezala, emaitza kaskarra-  
goak ematen ditu asmatze tasarekin konparatuta. Honek ere erlazio lineal bat jarraitzen du cache tamainarekin, baina kasu hontan, logikoki, alderantziz proportzionala da.

#### 4.4.3 Denbora Leihoaren arabera

Kasu hontan, bakarrik gure algoritmoaren eraginkortasuna azter dezakegu, beste bi algoritmoek denbora leihorik ez dutelako erabiltzen. Atal hontan, hiru denbora leiho ezarri ditugu, 300 segundukoa lehena, eta ondoren balioa bikoiztu eta hirukoiztu egingo dugu.

[4.9](#) irudiko grafikoan ikus daitekeenez, gero eta denbora leiho txikiagoa, gero eta emaitza hobeak ematen ditu. Denbora leihoaren portaera egonkortu egiten dela ondoriozta dezakegu behin 600 segunduko denbora leihoa erabiltzen dugunean.

Hala ere, esperimentu gehiago egin beharko lirateke ondorio zehatzen bat ateratzeko denbora leihoaren inguruan.



**4.9 Irudia:** *Emaitzak cache tamaina ezberdinetarako.*



## 5. KAPITULUA

---

### Difusioa

---

Atal honetan, proiektuan egindako difusio zientifikoa azalduko da. Bi atal izango ditu, lehenengoa maiatza hasieran idatzitako Extended Abstract bat, *European Conference on Networks and Communications*<sup>1</sup> (EuCNC) konferentziara bidali zena. Bigarren atalean konferentzia berdinerako hitzaldiaz hitz egingo da.

#### 5.1 Extended Abstract

Maiatza hasieran Marco Quartulli, Vicomtech-IK4-ko gainbegiralea, proposamen batekin etorri zitzaidan. Proiektuak hartu zuen norabidea ikusita, aproposa iruditzen zitzaien garatzen hasi ginen metodoa azaltzea extended abstract baten bitartez.

Artikulu honek, baldintza batzuk bete behar zituen. gehienez bi orri izan behar zituen, eta txantiloia espezifiko bat jarraitu behar zuen. Beraz momentuan nuen lana pixka bat aldera utzi eta idazteari ekin nion.

Maiatzak 13an bidali zen, deiaren *deadline* egunean. Konferentziako *Workshop* batera bidali zen, zehazki *Network Management, QoS and Security* izenburua zuena. Maiatzak 16-an, posta elektronikoa jaso nuen, nire proposamena onartua izan zela adieraziz.

A eranskinean atxikituta dago EuCNC konferentzian onartutako extended Abstract-a.

---

<sup>1</sup><http://www.eucnc.eu/>

## 5.2 EuCNC-n Hitzaldia

Nik eman beharreko hitzaldia, Sare eta komunikazioen Europar konferentziaren 25garren edizioaren barnean dago. Atenasen izango da Ekainaren 27tik 30ra. Nire hitzaldia, *Network Management, QoS and Security* izeneko *workshop* baten barruan izango da.

Hau da *workshop*-aren agenda<sup>2</sup>:

- 9.00 Introduction to the Network Management, QoS and Security Working Groups - Robert Mullins, Waterford Institute of Technology, Ireland
- 9.05 Keynote : "Network and cloud optimization: it's all about the model- Professor Danny Raz, Technion University, Israel
- 9.45 Paper Presentation : An Energy Efficient Architecture for 5G Network Management - Kieran Sullivan, Waterford Institute of Technology, Ireland
- 10.05 Paper Presentation : The SELFNET approach to Self Protection through Autonomic Network Management - Jose Alcaraz Calero, University of the West of Scotland
- 10.25 Coffee
- 10.35 Paper Presentation : NFV Reliability with Machine Learning - Marius Corici, Fraunhofer Fokus, Berlin
- 10.50 Paper Presentation : A New Radio Access Stratum Security Architecture Supporting Dynamic 5G Radio Access Networks – Peter Schneider, Nokia Bell Labs
- 11.10 Paper Presentation : Using Machine Learning to Detect Noisy Neighbours in 5G Networks - Bruno Ordozgoiti, Universitat Politècnica de Madrid
- 11.30 Paper Presentation : Towards Micro Segmentation in 5G Network Security - Olli Mammela, VTT Finland
- 11.50 Paper Presentation : Cooperative Caching in C-RAN using Bayesian Classification and Greedy Placement - Benat Azpiazu, VicomTech
- 12.05 Paper Presentation : A Monitoring Framework for Heterogeneous NFV/SDN Cloud Environments - Theodore Zahariadis, Synelixis

---

<sup>2</sup>[http://wiki.netmgmtwg.5g-ppp.eu/mediawiki/index.php/Workshop\\_agenda](http://wiki.netmgmtwg.5g-ppp.eu/mediawiki/index.php/Workshop_agenda)

- 12.25 Panel Discussion (20 minutes)
- 12.45 Finish



## 6. KAPITULUA

---

### Ondorioak

---

Atal honetan proiektuaren ondorioak analizatuko ditugu. Lehenik ondorio teknikoak azalduko dira, proiektuaren garapenarekin erlazio gehien dutenak, ondoren etorkizunerako lana, eta azkenik ondorio pertsonalak aurkeztuko dira, hau da, pertsonalki ikasitako hainbat lezio.

#### 6.1 Proiektuaren ondorioak

Emaitzetatik hainbat ondorio atera ditut. Aipagarriena, gure metodoak, emaitza hobekak lortu dituela asmatze tasetan eta zerbitzatzeko denboretan, kasu batzuetan, ez dira izan esperotakoak. Dударik gabe, cacheak modu kooperatiboan erabiltzearen ondorioetako bat izan liteke. Hala ere, emaitza hauek ikusita, gure metodoak sarearen trafikoa gutxitzen lagundu dezakeela frogatu dugu, asmatze tasa altuagoak lorturik, internetera edukien bila gutxiagotan joan behar delako.

Beste ondorio garrantzitsu bat, gure metodoaren azkartasuna da. Gure algoritmoak erabiltzen duen bayesiar sailkatzaile motari esker, eta honek sailkatzailea entrenatzeko eta ondoren estimatzeko duen azkartasunari esker, ia modu errektiboan lan egin dezakegu. Beraz, gure algoritmoaren helburu bat bete da, ahalik eta azkarrena izatea.

Aipatzekoa da baita ere, sortutako metodoa ongi baloratua egotea sareen eta komunikazioen ikerkuntza arloan, nik idatzitako artikulua konferentzia bateko workshop batean onartua izan delako.

Bukatzeko, aurreko ondorio guztiak kontuan hartuta, proiektuaren helburuak bete dira, nahiz eta gure lanari hainbat hobekuntza posible aurkitu ditugun, hurrengo atalean zehaztuko direnak.

## 6.2 Etorkizunerako lana

Etorkizunean lan honen inguruan egin daitezkeen hobekuntzen artean, hiru atal bereizi ditugu: sailkatzaileen arloan, kokapen algoritmoarekin erlazioatutakoak eta proba inguruneekin zerikusia dutenak.

### 6.2.1 Sailkatzailea

Ikasketa automatikoaren arloa oso zabala da, eta bide guztiak aztertzea ezinezkoa. Are eta gutxiago mota honetako proiektu batek duen iraupenarekin, jakinda exekuzioak denbora luzea behar dutela. Beraz, etorkizunerako, erabakitze sistema proaktiboagoak probatzea ondo egongo litzateke. Adibidez, zuhaitzak erabiltzen dituzten sailkatzaileak edo *Support Vector Machine* deituriko algoritmoak erabiltzea. Hauek, aurredefinituriko periodoetan entrenatuko lirateke, adibidez, 30 minuturo.

Bestalde, sailkatzaileari atributu gehiago pasatuko nizkioke. Lan honetarako *ground truth* faltagatik, ezin izan dira gehitu erabiltzaileen datuak, hauek beti oso baliagarriak dira edukien eskaerak iragartzeko. Youtube-k adibidez, behin bideo bat amaituta, erabiltzaileen datuen arabera eduki bat gomendatzen du, eta gure metodoarentzat egokia litzateke honelako aurreikuspen sistema bat gehitzea.

Baita ere, eduki batzuk urtarokotasun batekin eskatzen dira, eta urtarokotasun hauek ere detektatzea garrantzitsua izan daiteke, hain zuzen ere sailkatzailean aplikatzeko atributu izan litekeelako. Saiakera bat egin zen, Holt-Winters metodoarekin, abstract-ean gehitua zegoena, baina atzera bota zen emaitza kaskarrak zituelako. Mota honetako hainbat algoritmo erabili daitezke, adibidez gaur egun oso erabilia den *Autoregressive Integrated Moving Average* (ARIMA) izeneko metodo bat.

### 6.2.2 Kokapena

Kokapenari dagokionez, hobekuntza nabarmen bat, gure metodoaren emaitzak ikusita, kokapen metodoari edukien errepikapenaren posibilitatea gehitzea da, zeren ateratako

ondorio nagusi bat, cache hit ratioa hobetu dugu, baina batzuetan zerbitzatzeko denbora kaskarragoa lortu da. Beraz, emaitza hauek hobetzeko helburuarekin eduki ospetsuenak cache ezberdinetan errepikatzea izango litzateke soluzio bat.

Gur erabilitako kokapen funtzioak aurreko denbora leiho bat erabiltzen du kokapen optimoa aukeratzeko. Beraz, iragana erabili beharrean, etorkizuna erabiltzea agian emaitzak hobetuko litzateke. Hortarako, lehen aipaturako ARIMA algoritmoa egokia izango litzateke, edo denbora serieetan oinarritutako metodoen bat.

### 6.2.3 Proba Ingurunea

Gure proba ingurunea, nik sortutako simulatzaile bat izan da. Simulatzaile hau ahalik eta errealistena izaten saiatu da. Ahalik eta datu gehien jasotzen saiatu gara simulatzaile hau sortzeko. Hala ere, datu faltaren ondorioz, ezin izan dira beste funtzio batzuk gehitu. Etorkizunerako datu hauek lortzea eta simulatzailean integratzea ondo egongo litzateke. Adibidez, lehen aipaturiko erabiltzaileen dataset bat erabiltzea ondo egongo litzateke, edo eskaeren dataset bat.

Simulatzaile bat erabili beharrean, sistema erreal batean aplikatzea bikaina izango zen. Azkenean, nahiz eta simulazioak ahalik eta errealistenak egin, beti egongo da errealitateik pixka bat aldentuta. C-RAN arkitektura duen, eta cacheak dituzten ingurune batean probak egitea interesgarria izango litzateke.

## 6.3 Ondorio pertsonalak

Motibazioetan azaldu bezala, lan hau maila akademikotik lan mundura saltoa egitean datza. Nahiz eta ni 2 urtez garatzaile bezala lanean ibili izan, unibertsitatera sartu aurretik, bi eredu ezberdinez hitz egiten ari gara, bat enpresa, eta oraingoa, Vicomtech-IK4, zentro teknologiko bat. Bien helburuak oso ezberdinak dira funtsean, baina errealitatean antzerako ezaugarriak dituzte. Proiektu motak oso ezberdinak dira batean eta bestean.

Zentro teknologikoarekin zer ikusia duena baita ere, Ikerkuntza, Garapen eta berrikuntza proiektu batean parte hartzean datza, gainera europar proiektu batean. Nahiz eta ni ez egon beste kideekin bileretan, posta bidez proiektuaren nondik norakoetaz jabetzen nintzela. Orain arte mundu akademikoan oso gauza txikiak egin ditugu ikerkuntzarekin erlazionatuta, hemen asko ikasi dut, adibidez, artikuluko zientifikoen bilaketa eta filtraketan, edota artikuluen sailkapenean.

Sareen munduari buruz ere asko ikasi behar izan dut, konputazio espezialitatetik joanda, sareei buruzko irakasgai bakarra eman dudalako graduan zehar, eta sareen oinarriei buruzkoa zen. Beraz, hemen kurtso azeleratu bat behar izan dut gaur egun dauden sareei buruz.

Horrekin ez kontentu, artikulua bat idatzi behar izan dut, horrek dakarren lan guztiarekin. Hasieran erraza izango zela uste nuen, baina azkenik ez da horrela izan. Sintesi gaitasun handia izan behar da, eta guztia lengoai konkretu batekin, normalean artikulua asko irakurri eta idatzi dituzunean kontrolatzen hasten zarena.

Ikerkuntzarekin jarraituz, artikulua hori europar konferentzia batean onartua izateak ere bere ondorioak izan ditu, Atenasera joatea hitzaldi bat ematera, memoria hau idazten ari naizen momentuan oraindik gertatu ez dena, beraz, ezin dira mamitu ikasi daitezkeen lezio guztiak, baina seguruaski asko izango direla, berria baita niretzat.

Kudeaketaren aldetik, esan beharra dago *Proiektuen Kudeaketa* irakasgaiaren ikasitako hainbat kontzeptu erabiltzeak proiektua errazago aurrera eramaten lagundu didala. Proiektuen atazak ondo zehaztu eta hauentzako datak zehazteak benetan laguntzen du proiektuaren garapenaren kontrola egin ahal izateko.

Arriskuen kudeaketak ere bere garrantzia izan du, aurrez eginiko identifikazioak eta aurremateak garrantzi handia izan du. Arazoren bat sortu zaidan bakoitzean ez du proiektuan eragin handirik izan.

Diseinu aldetik, hasieran sistema ondo diseinatzeak bere garrantzia ere izan du. Aldaketak egiterako orduan, puntu askotan kodea atzera ez botatzeak denbora eta lan asko aurreztu dit.



# **Eranskinak**



## **A. ERANSKINA**

---

### **Extended Abstract-a**

---

Eranksin honetan, EuCNC konferentziara bidalitako extended abstract-a azaltzen da. Maiztza hasiera idatzi zen, beraz azaltzen diren gauza batzuk kentzea erabaki da beranduago, sistema hobetzeko xedearekin.

# Cooperative Caching in C-RAN using Bayesian Classification and Greedy Placement

Beñat Azpiazu, Marco Quartulli, Angel Martin,  
Igor G. Olaizola  
Digital Media  
Vicomtech-IK4  
San Sebastian, Spain  
bazpiazu@vicomtech.org, mquartull@vicomtech.org,  
amartin@vicomtech.org, iolaizola@vicomtech.org

Basilio Sierra  
Computer Sciences and Artificial Intelligence Department  
University of the Basque Country  
San Sebastian, Spain  
b.sierra@ehu.eus

**Abstract**— Machine Learning and statistical forecasting algorithms for the autonomic management of 5G networks need to be evaluated by well-defined quantitative means. We propose setting up a simulation-based environment for such evaluation, using it to quantify the performance of the setup proposed by [1] for distributed caching and to subsequently improve upon that go using on ML-based popularity estimation and greedy algorithms for a smart content distribution among the different caches.

**Keywords**—caching, machine learning, Bayes, greedy, placement, cloud, RAN, cooperative.

## I. INTRODUCTION

To enable innovation in domains as diverse as manufacturing, energy, health and media [2], 5G telecommunication networks will need to improve network throughput of at least an order of magnitude. Smart distribution and coordination of caching functionality among the nodes at the edge of the network has been demonstrated [1] to have the potential to contribute to realizing this objective, allowing for more rapid delivery and minimizing backhaul traffic.

## II. ARCHITECTURE - CLOUD-RADIO ACCESS NETWORK

As explained in [1], a Base Band Unit (BBU) cloud Central Cache Manager (CCM) monitors user requests from the local Cloud Radio Area Network (C-RAN) and is responsible to make cache placement decisions.

While the actual content files are physically stored in separated caches, a global indexing table can be maintained by the CCM to facilitate content lookup and cache management. When users in a cell make requests for some file already stored in the edge cache at Radio Remote Head (RRH), they can directly download it from a local RRH without incurring fronthaul traffic. If the requested file is not stored in the local edge cache, the request is forwarded to the CCM in the BBU cloud, which will search for the file in the cloud cache and in all the neighboring edge caches. If CCM cannot locate the requested file, it will direct the request to the original content server in the remote Content Distribution Network (CDN), incurring traffic in the backhaul links.

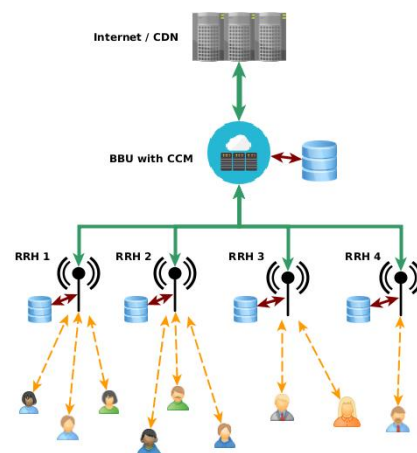


Figure 1: Illustration of the abstraction of a C-RAN

## III. METHOD

In this section, we consider using Machine Learning and greedy algorithms to solve the optimal cache placement problem. Firstly we estimate the future amount of requests by using a Holt-Winters model [3], commonly used for forecasting in financial analytics.

After that, we use files common metadata and the previous prediction with a Naïve Bayes classifier, to explicitly estimate the probability of getting a request in the future.

With these probabilities, we apply a greedy algorithm to place files in the optimal cache of the network, so that delivery delay is minimized.

### A. Holt-Winters request prediction

Holt-Winters model is an exponential smoothing technique used to predict future requests in a given period. Unlike many other techniques, this model can easily adapt to changes, trends and stationary changes.

We estimate the future amount of requests for each file in a period of time extrapolating current data using a Holt-

Winters model. Then we add the estimations as attributes to our machine learning techniques.

### B. Future request estimation using Naïve Bayes

Naïve Bayes is one of the most commonly used classifiers because of its simplicity and speed. It is based on Bayes theorem, also known as the conditional probability theorem.

We use the Naïve Bayes model to calculate the probability of reception of a request in a well- defined future time window. For our predictions, we use this set of attributes:

Attribute	Meaning
$X_1$	Time since last request
$X_2$	Frequency of the requests in a limited time window
$X_3$	Frequency of the request in an extended time window
$X_4$	Size of the file
$X_5$	Holt-Winters prediction for the request rate

We apply Laplace smoothing techniques to continuous attributes. As target we set a binary label, 0 if the request will most likely not repeat in the future and 1 if the request will most likely repeat. In our case, the formula for Naïve Bayes is the following:

$$p(C = 1 | X_1 = x_1, \dots, X_n = x_n) = p(C = 1) * \prod_{i=1}^n p(X_i = x_i | C = 1)$$

### C. Greedy algorithm for placing problem

We apply a greedy algorithm to place the files in the most effective cache. The greedy algorithm is based on the cost to reach the file from previous requests and the size.

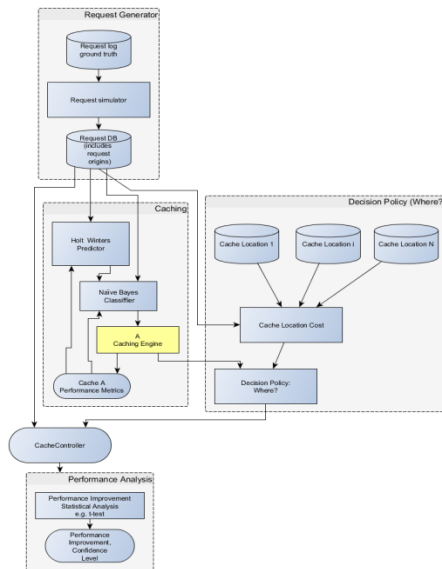


Figure 2: Diagram of the whole method to classify files by popularity and cache them in into the best cache possible in the network

## IV. TEST ENVIRONMENT

To reach the goals described before, we carry out discrete simulations of the operation of a system that mimics the approach described in the Section II. To define the discrete request generator, we consider a Zipf distribution to generate popularity-based request, as in [4]. We use a real dataset [5] for file size distribution densities.

The simulation returns the following outputs: Average cache hit ratio, average Delivery delay, graphic with the distribution of Delivery delay.



Figure 3: Simulation output graphic example (Delivery delay PDF), with LRU algorithm and LCE placing algorithm.

## CONCLUSIONS

Machine Learning and statistical forecasting algorithms for cooperative caching in 5G networks are useful, and need to be evaluated by well-defined quantitative means.

For future research, we can investigate future request predictions with other models as Autoregressive Moving Average models (ARMA). We will also take into account the backhaul traffic congestion in the simulation, to optimize cache placements.

## ACKNOWLEDGEMENT

This work was supported by the EU project CogNet, 671625 (H2020-ICT-2014-2, Research and Innovation action).

## REFERENCES

1. Wu, J., Zhang, Z., Hong, Y., & Wen, Y. (2015). Cloud radio access network (C-RAN): a primer. *Network, IEEE*, 29(1), 35-41.
2. 5G-PPP, ERTICO, EFFRA, EUTC, NEM, CONTINUA and Network2020 ETP (2016). 5G empowering vertical industries
3. Kalekar, P. S. (2004). Time series forecasting using holt-winters exponential smoothing. *Kanwal Rekhi School of Information Technology*, 4329008, 1-13.
4. L. Breslau, Pei Cao, Li Fan, G. Phillips and S. Shenker, "Web caching and Zipf-like distributions: evidence and implications," *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, New York, NY, 1999, pp. 126-134 vol.1.
5. Mattia Zeni, Daniele Miorandi, and Francesco De Pellegrini. 2013. YOUStatAnalyzer: a tool for analysing the dynamics of YouTube content popularity. In *Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools (ValueTools '13)*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 286-289.



---

## Bibliografia

---

- [AD14] H. Ahleghagh and S. Dey. Video-aware scheduling and caching in the radio access network. *IEEE/ACM Transactions on Networking*, 22(5):1444–1462, Oct 2014.
- [BBD14] E. Bastug, M. Bennis, and M. Debbah. Living on the edge: The role of proactive caching in 5g wireless networks. *IEEE Communications Magazine*, 52(8):82–89, Aug 2014.
- [BCF<sup>+</sup>99] L. Breslau, Pei Cao, Li Fan, G. Phillips, and S. Shenker. Web caching and zipf-like distributions: evidence and implications. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 126–134 vol.1, Mar 1999.
- [BGW10] S. Borst, V. Gupta, and A. Walid. Distributed caching algorithms for content distribution networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, March 2010.
- [GKJA15] Ammar Gharaibeh, Abdallah Khreishah, Bo Ji, and Moussa Ayyash. A provably efficient online collaborative caching algorithm for multicell-coordinated systems. *CoRR*, abs/1509.02911, 2015.
- [GSD<sup>+</sup>12] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire. Femtocaching: Wireless video content delivery through distributed caching helpers. In *INFOCOM, 2012 Proceedings IEEE*, pages 1107–1115, March 2012.
- [SD14] Yasir Shoaib and Olivia Das. Performance-oriented cloud provisioning: Taxonomy and survey. *CoRR*, abs/1411.5077, 2014.

- [THP16] Tuyen X. Tran, Abolfazl Hajisami, and Dario Pompili. Cooperative hierarchical caching in 5g cloud radio access networks (c-rans). *CoRR*, abs/1602.02178, 2016.
- [WCT<sup>+</sup>14] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung. Cache in the air: exploiting content caching and delivery techniques for 5g systems. *IEEE Communications Magazine*, 52(2):131–139, February 2014.