

UNIVERSITETI I PRISHTINËS “HASAN PRISHTINA”
FAKULTETI I INXHINIERISË ELEKTRIKE DHE KOMPJUTERIKE



PROJEKTI: DIZAJNIMI KLIENT-SERVER
LËNDA: RRJETAT KOMPJUTERIKE

Mentor: Prof. Dr. Blerim Rexha
MSc. Haxhi Lajqi

Punoi: Arlinda Kastrati
ID: 190714100006

Prishtinë, 2021

Abstrakti

Ky raport paraqet një përmbledhje të detajeve në dizajnimin, implementimi dhe testimi i programit klient dhe server në versionet TCP dhe UDP .Ky projekt është implementuar duke përdorur soketa dhe gjuhën programuese Python.

Objektivi kryesor i këtij raporti është të paraqesë parimet që qëndrojnë prapa programimit të programit komunikues klient/server përmes soketeve në TCP/UDP.

Permbajtja

Abstrakti	2
Hyrja	4
1. Metodat	5
1.1 Metodat kryesore	5
Metoda IP	5
Metoda NRPORTIT	5
Metoda NUMERO	5
Metoda ANASJELLTAS	6
Metoda PALINDROM	6
Metoda KOHA	6
Metoda LOJA	6
Metoda GCF	7
Metoda KONVERTO	7
1.2 Metodat shitesë	8
Metoda ANAGRAM	8
Metoda PRIM	8
2. FIEK TCP –PROGRAMIMI PERMES SOKETEVERE	9
2.1 TCP	9
2.2 TCP-Modeli i komunikimit përmes soketave Klient/Server	10
2.3 TCP Klienti	11
2.4 TCP-Serveri	12
3. FIEK UDP –PROGRAMIMI PERMES SOKETEVERE	13
3.1 UDP-Protokolli i te Dhenave te Perdoruesit	13
3.2 UDP-Modeli i komunikimit përmes soketave Klient/Server	13
3.3 UDP-Klienti	14
3.4 UDP-Serveri	15
4. Testimi i programit klient/server ne TCP dhe UDP	16
5. Perfundimi	18
6. Referencat	19

Hyrja

Interneti është pa dyshim sistemi më i madh i inxhinierisë i cili është krijuar ndonjëherë nga njerëzimi, me qindra miliona kompjutera të lidhur me vete përmes lidhjeve komunikuese gjegjësisht rrjetave. Në të kaluarën, rrjetet kompjuterike ishin një grup kompjuterash të ngjashëm, të lidhur me një numër të madh të kabllorëve dhe të palevizshëm. Në ditët e sotme, rrjetet kompjuterike janë shumë fleksibile. Teknologjia e rrjetave është ajo e cila lejon komunikimin e miliarda përdoruesve nga e gjithë bota të cilët lidhen përmes laptopëve, tabletëve dhe telefonave të mençur.

Një rrjetë është e përbërë nga kompjuterë i cili është një klient ose një server. Një server është një program që ofron disa shërbime, ndërsa klienti është një program që është duke kërkuar disa shërbime.

Serverët janë kompjuterë të fuqishëm ose procese të dedikuara për menaxhimin e disqeve të diskut (skedari serverat), printerët (serverat e printimit) ose trafiku i rrjetit (shërbimet e rrjetit) ndërsa **klientët** janë PC ose stacione pune në të cilën përdoruesit ekzekutojnë aplikacione.

Kur një përdorues kërkon një shërbim, pajisja e tij duhet të jetë e lidhur në një rrjetë, qoftë ajo rrjetë lokale ose internet. Kështu që kur një pajisjeje i nevojitet lidhja në rrjetin lokal apo në internet, ajo pajisje përdorë komponenten softwerike të quajtur **soket**.

Soketi është një pikë fundore e një lidhje komunikimi dykahëshe midis dy programeve në një rrjetë. Soketet lejojnë komunikimin midis dy proceseve të ndryshme në të njëjtën ose në pajisje të ndryshme.

Lejimi i këtij komunikimi është i mundur përmes protokolleve siq janë **TCP/IP** dhe **UDP**.

Me poshtë kemi dizjnimin, implementimin dhe testimin e një programi klient/server të implementuar përmes protokolleve TCP/IP dhe UDP.

Veglat softwerike për realizimin e projektit:

- Sistemi Operativ: Microsoft Windows 10
- Gjuha programuese dhe version i saj: Python 3.9.2
- Editori: Visual Studio Code, Visual Studio 2020

1. Metodat

Të gjitha metodat janë implementuar përmes funksioneve pra me fjalën kyqe **def**.

1.1 Metodat kryesore

Metoda IP

Përcakton dhe kthen IP adresën e klientit përmes **socket.gethostbyname(serverName)** por është e mundur edhe të realizohet përmes **str(address[0])**.

```
def IP():  
    return socket.gethostbyname(serverName)
```

Metoda NRPORTIT

Përcakton dhe kthen portin e klientit përmes **str(address[1])**. Pas deklarimit të IP adresës me emrin **serverName** dhe të portit me emrin **serverPort**, gjatë krijimi të soketave këta dy parametra konsiderohen si **address=(serverName,serverPort)** by default. Andaj **str(address[1])** shfaq numrin e portit në string.

```
def NRPORTIT():  
    return str(address[1])
```

Metoda NUMERO

Përcakton numri e zanoreve dhe bashkëtingelloreve të një teksti të dhënë. Deklarimi i variablave numëruese **zanoret=0** dhe **bashkëtingellore=0**, për çdo karakter në një rang prej 0 deri te **gjatësia e teksiti** të dhënë duke krahasuar nëse ai karakter gjendet në **listen (a,e, ë,I,o,u,y,A,E, Ē,I,O,U,Y)** nëse gjendet numëruesin e zanoreve **rrite +1**, përndryshe **rrite te bashkëtingelloreve** dhe kthe nr e tyre.

```
def NUMERO(Teksti):  
    teksti=str(Teksti)  
    zanoret=0  
    bashkëtingellore=0  
    listazanoreve =('a','e','ë','i','o','u','y','A','E','Ē','I','O','U','Y' )  
    listabashkëtingelloreve =('b','c','ç','d','dh','f','g','gj','h','j','k','l','ll','m','n','nj','p','  
        'B','C','Ç','D','Dh','F','G','Gj','H','J','K','L','Ll','M','N','Nj','P','Q','R','Rr','S','Sh'  
    for i in range(0,len(Teksti)):  
        if Teksti[i] in listazanoreve:  
            zanoret=zanoret+1;#numëruesi i zanoreve rritet per 1  
        elif Teksti[i] in listabashkëtingelloreve :  
            bashkëtingellore=bashkëtingellore+1 #numëruesi i bashkëtingelloreve rritet per 1  
    x=str(zanoret)  
    z=str(bashkëtingellore)  
    return str("Teksti ka "+x+" zanore dhe "+z+" bashkëtingellore.")
```

Metoda **ANASJELLTAS**

Kthen tekstin që e ka marr si parameter në fillim në anën e kundërt. **Teksti[::-1]** nënkupton se do të fillosh nga fundi i stringut dhe të përfundosh në pozicioni **0** pra pozicionin e parë ,me hapin **-1** që do të thotë një hap prapa.

```
def ANASJELLTAS(Teksti):  
    return Teksti[::-1]
```

Metoda **PALINDROM**

Palindromi nuk është gjë tjetër veçse ndonjë numër ose fjali e cila mbetet e pandryshuar edhe kur përmbysset. Përmes kushti **if Teksti==Teksti[::-1]** krahason nese Teksti i marr si parameter është i njejte me Tekstin e permbysur (Metoda ANASJELLTAS). Dhe përmes **return("Teksti...")** na tregon nese fjalia është palindrome apo jo.

```
def PALINDROM(Teksti):  
    if Teksti==Teksti[::-1]:  
        return ("Teksti i dhene eshte palindrome.")  
    else:  
        return("Teksti i dhene nuk eshte palindrome.")
```

Metoda **KOHA**

Përcakton kohën aktuale në server dhe e dërgon atë tek klienti varësisht formatit që ai e ka zgjedhur.Për realizimin e saj nevojitet importimi i libraris datetime.

```
def KOHA():  
    kohatani=datetime.now()  
    return (kohatani.strftime("%m.%d.%Y %H:%M:%S %p") )
```

Metoda **LOJA**

Ka për qëllim që të kthej 5 numra nga rangun i caktuar [1,35]. **Rendomlista = random.sample(range(1,35),5)** gjenron nje varg prej 5 numrash në rangun 1-35, deklarojme një varg të zbrazët ku vendosen numrat pas sortimit dhe kthejme atë.

```
def LOJA():
    randomlista = random.sample(range(1, 35), 5)
    listasortuar = []
    listasortuar = sorted(randomlista)
    nrsortuar=str(listasortuar)
    return str('p.sh ' +nrsortuar+' 5 numra te rastesishem nga 35.')
```

Metoda GCF

Përcakton faktorin më të madh të përbashkët në mes dy numrave dhe kthen atë që është. Shfrytëzojmë funksionin që gjenden tek libraria **math** sic është **math.gcd**.

```
def GCF (Nr1,Nr2):
    x=math.gcd(Nr1,Nr2)
    return str(x)
```

Metoda KONVERTO

Kthen si rezultat konvertimin e një vlerë në një vlerë tjetër varësisht opcionit të zgjedhur. Opcionet e mundshme janë: **cmNeInch inchNeCm kmNeMiles mileNeKm**.

```
def KONVERTO(opcioni,Numri):
    if opcioni=='cmNeInch':
        rez = Numri/2.54
    elif opcioni=='inchNeCm':
        rez = 2.54*Numri
    elif opcioni=='kmNeMiles':
        rez = Numri*0.621371
    elif opcioni == 'mileNeKm':
        rez = Numri*1.60934
    else:
        rez='Gabim'
    return str(round(rez,2))
```

1.2 Metodat shitesë

Metoda **ANAGRAM**

Përcakton nëse dy tekste janë anagrame. Anagram-nënkupton nëse dy tekste të ndryshme me gjatësi të njëjtë përmbajnë shkronjat e njëjta.

```
def ANAGRAM(Teksti1,Teksti2):
    Teksti1=Teksti1.lower()
    Teksti2=Teksti2.lower()
    if(len(Teksti1)==len(Teksti2)):
        sortimiTeksti1=sorted(Teksti1)
        sortimiTeksti2=sorted(Teksti2)
        if(sortimiTeksti1 == sortimiTeksti2):
            return (Teksti1 + " dhe "+Teksti2+" jane anagram.")
        else:
            return (Teksti1 + " dhe "+Teksti2+" nuk jane anagram.")
    else:
        return (Teksti1 + " dhe "+Teksti2+" nuk jane anagram.")
```

Metoda **PRIM**

Përcakton nëse një numer është i thjesht dhe kthen përgjigjen në fomen booleane.

```
def PRIM(Num1):
    if Num1 <= 1 or Num1 % 1 > 0:
        return False
    for i in range(2, Num1//2):
        if Num1 % i == 0:
            return False
    return True
```


2. FIEK TCP –PROGRAMIMI PERMES SOKETEVERE

2.1 TCP

TCP(Transmission Control Protocol- Protokolli i Kontrollit të Transmetimit) është një standard i komunikimit që mundëson programet dhe pajisje kompjuterike për të shkëmbyer mesazhe mes vete në një rrjet.

Pasi që,TCP është e orientuar drejt lidhjes(connection-oriented) ,fillimisht duhet vendosur një lidhje midis klientit dhe serverit përpara se të dërgohen të dhënat. Serveri duhet të jetë duke dëgjuar për kërkesat e lidhjes nga klientët para se të vendoset lidhja ose te behet connection me klientin.

TCP është i përshtatshëm për aplikacione apo programet që kërkojnë besueshmëri të lartë dhe koha e transmetimit është relativisht jo shume kritike. Përdoret nga protokollet si HTTP, FTP, SMTP etj.

TCP rregullon paketat e të dhënave në rendin e specifikuar. Njëkohesisht,ekziston një garanci absolute se të dhënat e transferuara mbeten të paprekura dhe arrijnë me të njëjtin rend me të cilin janë dërguar. TCP merret me besueshmërinë,kontrollin e rrjedhes se transmetimit se te dhenave, kontrollin e mbingarkesës, kontrollimin e gabimeve dhe rikuperimin e gabimeve. Paketat e gabuara ritransmetohen nga burimi në destinacion

IP(Internet Protocol-Protokolli I Internetit) është metoda për dërgimin e të dhënave nga një pajisje në tjetrën përmes internetit. Çdo pajisje ka një adresë IP që e identifikon atë në mënyrë unike dhe i mundëson asaj të komunikojë dhe të shkëmbejë të dhëna me pajisje të tjera të lidhura në internet.

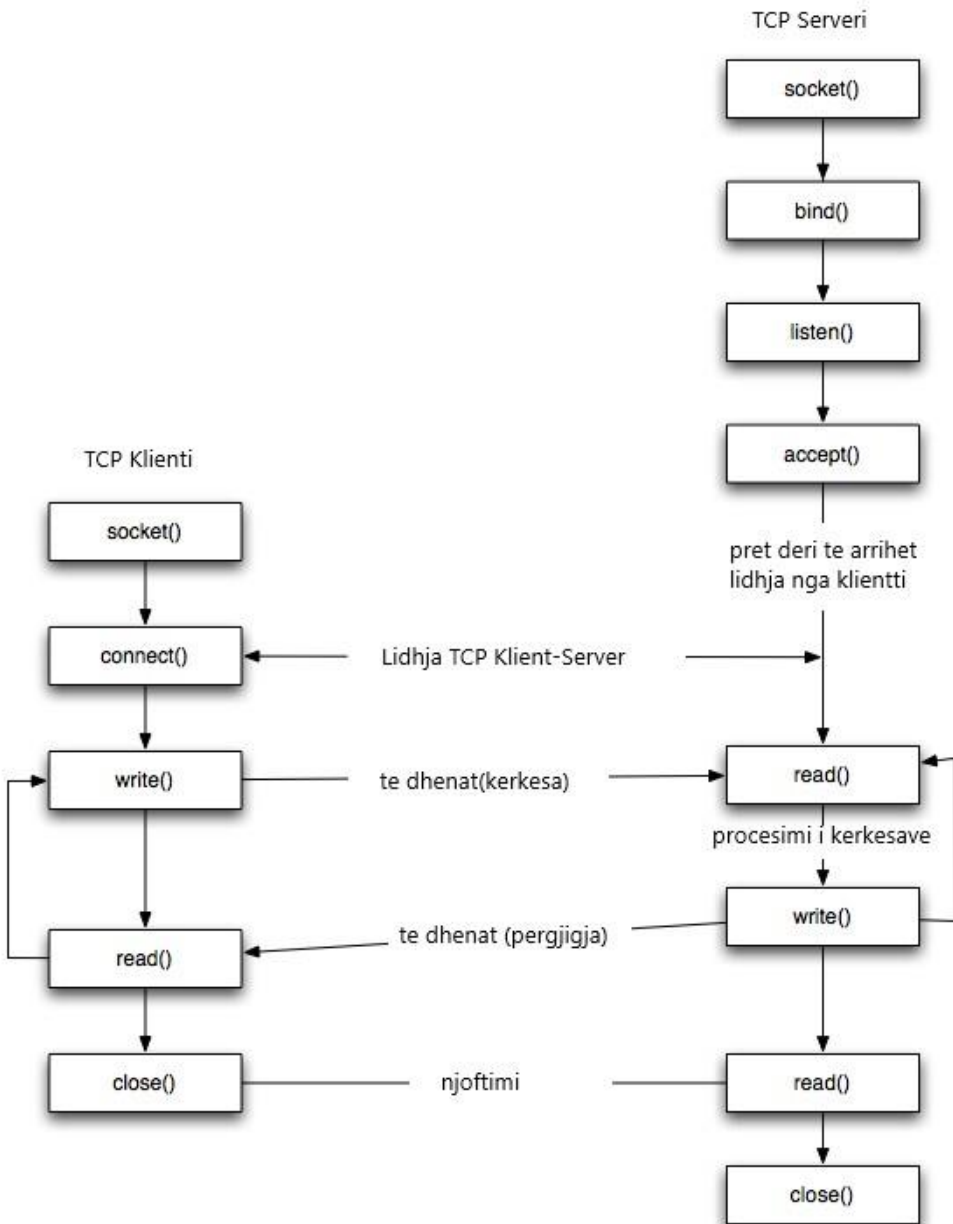


127.0.0.1
172.16.0.9
192.0.0.7

TCP dhe IP janë protokolle të veçanta që punojnë së bashku për të siguruar që të dhënat dorëzohen në destinacionin që eshte synuar ne fillim brenda nje rrjeti.

IP merr dhe përcakton adresën e aplikacionit ose pajisjes në të cilën duhet të dërgohen të dhënat. TCP më pas është përgjegjës për transportimin e të dhënave dhe sigurimin që ato dorëzohen në aplikacionin e destinacionit ose pajisjen që IP ka përcaktuar.

2.2 TCP-Modeli i komunikimit përmes soketave Klient/Server



- 1.Realizimi i lidhjes TCP Klienti-Serveri
- 2.Klienti lexon kërkesat nga tastiera dhe i dergon tek serveri
- 3.Serveri lexon kërkesen e marr nga Klienti dhe e proceson varësisht se qfarë kërkon kërkesa
- 4.Serveri dërgon përgjigjen e kërkesës
- 5.Klienti e lexon përgjigjen e pranuar
- 6.Klienti mbyllet dhe dergon njoftimin te severi
- 7.Serveri lexon njoftimin dhe mbyllet

2.3 TCP Klienti

Kodi për implementimin e soketit tek TCP-Klienti. Kodi i plotë gjendet ne fjallin TCP-Klienti.py.

```
import socket #Importi i te gjitha moduleve nga libraria socket

serverName = '127.0.0.1' #IP Adresa e localhostit
serverPort = 14000 #NUMRI I PORTIT te serverit
#Funksioni/Kushtet per nderrim te portit
#.....

soketi = socket.socket(socket.AF_INET, socket.SOCK_STREAM)#Ky rresht krijon soket
in e klientit te quajtur soketi dhe permban dy parametra.
#Parametri i pare AF_INET-tregon se rrjeti perdore IPv4
#Parametri i dyte SOCK_STREAM-
tregon se soketi eshte i tipit SOCK STREAM qe nenkupton qe oshte TCP soket

soketi.connect((serverName,serverPort))#Komunikimi mes Klientit dhe Serverit
#Parametri i metodes connect eshte adresa e anes se komunikimit te Serverit

print('-----SOKETI TCP_Klienti-----')
kerkesa = input("I/e nderuara klient/e ju sapo jeni lidhur me serverin,ju lutem s
hkruani kerkesen tuaj: ")
#.....
while True:#unaza e pafundme
    if (kerkesa == ""): #Nese kerkesa eshte ENTER
        soketi.close() #Soketi do te mbyllet
        print('Lidhja me serverin eshte mbyllur') #Lajmerimi qe lidhja me
        serverin rin eshte mbyllur
        break
    else:
        #Dergimi i kerkeses tek serveri(gjegjesishte ne localhost me portin
        #qe e kemi percaktuar tek serverPort)

        soketi.sendall(kerkesa.encode())#Default encoding=utf-8
        #Variabla pergjigja nenkupton pergjigjen nga serverin ne baze te
        kerkesave te parashtruara
        pergjigjja = soketi.recv(128).decode()#Default decoding=utf-8
        print(pergjigjja) #Printimi i pergjigjes
        kerkesa = input('Ju lutem shkruani kerkesen tuaj: ')

soketi.close()
```

2.4 TCP-Serveri

Kodi për implementimin e sokerit tek TCP-Serveri. Kodi i plotë gjendet në fjallin TCP-Serveri.py.

```
#Importimi i librarive te nevojshme
import socket
#.....
import threading

serverName = '127.0.0.1' #IP Adresa e localhostit
serverPort = 14000 #NUMRI I PORTIT te serverit

serverS = socket.socket(socket.AF_INET,socket.SOCK_STREAM)#Ky rresht krijon soker
in e serverit te quajtur serverS dhe permban dy parametra.
#Parametri i pare AF_INET-tregon se rrjeti perdore IPv4
#Parametri i dyte SOCK_STREAM-
tregon se sokeri eshte i tipit SOCK STREAM qe nenkupton qe oshte TCP soker
serverS.bind((serverName, serverPort)) #Percaktimi i serverName dhe serverPort pe
r lidhje
print('Serveri eshte startuar ne localhost ne portin ' + str(serverPort))
serverS.listen(5) #Serveri pret kerkesat e klientit
print('-----TCP_Serveri-----')
print('Server eshte duke punuar dhe eshte duke pritur per ndonje kerkesa!')
#Funksionet
#.....
#Funksionet
#Threads
#...
while True:
    print('-----')
    connection, address = serverS.accept() #Pergjigjja,pranimi i nje lidhjeje te
    re ne sokerin e serverS
    print("Serveri është lidhur me klientin me IP Adresë %s, në portin %s" % addr
ess)
    threading._start_new_thread(ThreadFunction,(connection,)) #Krijimi i threads
serverS.close()
```

3. FIEK UDP –PROGRAMIMI PERMES SOKETEVERE

3.1 UDP-Protokolli i te Dhenave te Perdoruesit

UDP(User Datagram Protocol-Protokolli i Datagram e Perdoruesit) është një nga protokollet kryesore në grupin e protokolleve të internetit. Me UDP, programet dhe pajisjet kompjuterike mund të dërgojnë mesazhe, të referuara si të datagrame(dhena), të hostet e tjera që janë të identifikuar me IP.

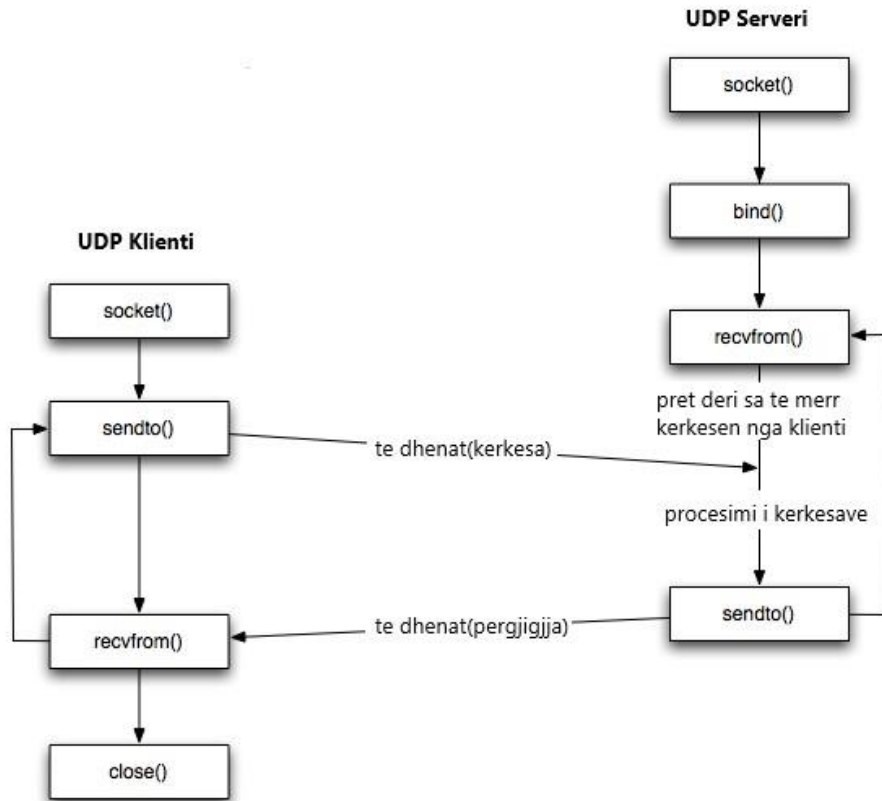
Përdoret, posaçërisht për aplikacione të ndjeshme si lojërat, luajtja e videove ose kërkimet e Sistemit të Emrit të Domenit (DNS). UDP rezulton në komunikim më të shpejtë sepse nuk kalon kohë duke formuar një lidhje të fortë me destinacionin para transferimit të të dhënave sikurs TCP.

UDP vjen me kontrolle, të cilat kanë për qëllim të sigurojnë integritetin e të dhënave dhe numrat e portave, të cilët ndihmojnë në dallimin e rolit që luajnë të dhënat në burim dhe destinacion.

Ne krahasim me TCP ka disa të meta sepse pas identifikimit të një pajisjeje edhe pas dërgimit të të dhënave, nuk ekziston diqka e tillë për të treguar rendin me të cilin duhet të mbërrijnë të dhënat apo edhe një njoftim nëse të dhënat arriten në destinacionin e caktuar në fillim.

Për këtë arsye edhe është e ekspozuar ndaj rrjeteve të ndryshme dhe sulmohet.

3.2 UDP-Modeli i komunikimit përmes soketave Klient/Server



1. Serveri pret deri sa të marrë kërkesën nga klienti
2. Klienti lexon kërkesat nga tastiera dhe e dërgon te serveri
3. Serveri lexon kërkesën e marrë nga Klienti dhe e proceson varësisht se çfarë kërkon kërkesa
4. Serveri dërgon përgjigjen e kërkesës

- 5.Klienti e lexon përgjigjen e pranuar
- 6.Klienti mbyllet

3.3 UDP-Klienti

Kodi për implementimin e soketit tek UDP-Klienti. Kodi i plotë gjendet në fjallin UDP-Klienti.py.

```
#Programimi i soketit FIEK Klienti permes protokollit TCP
import socket #Importi i te gjitha moduleve nga libraria socket

server_name = '127.0.0.1'#IP Adresa e localhostit
server_port = 14000 #NUMRI I PORTIT te serverit
#Funksioni/Kushtet per nderrim te portit
#.....
soketiK = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)#Ky rresht krijon soketi
n e klientit te quajtur soketiK dhe permban dy parametra.
#Parametri i pare AF_INET-tregon se rrjeti perdore IPv4
#Parametri i dyte SOCK_DGRAM-
tregon se soketi eshte i tipit SOCK DGRAM qe nenkupton qe oshte UDP soket

print('-----SOKETI UDP_Klienti-----')
#.....
kerkesa = input("I/e nderuara klient/e ju sapo jeni lidhur me serverin,ju lutem s
hkruani kerkesen tuaj: ")

while True:
    #Dergimi i kerkeses tek serveri(gjegjesishte ne localhost me portin qe e kemi
    percaktuare tek serverPort)
    soketiK.sendto(kerkesa.encode(), (server_name, server_port))
    pergjigjja = soketiK.recv(128).decode()#Default decoding=utf-8
    print(pergjigjja) #Printimi i pergjigjes

soketiK.close()
```

3.4 UDP-Serveri

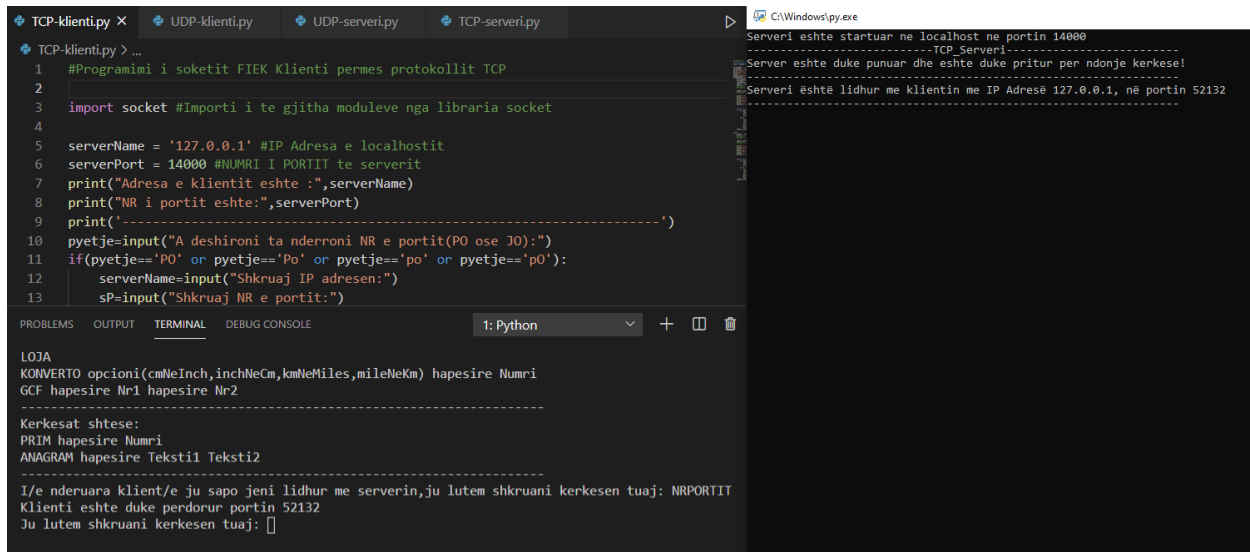
Kodi për implementimin e sokitit tek UDP-Serveri. Kodi i plotë gjendet në fjallin UDP-Serveri.py.

```
#Importimi i librarive te nevojshme
import socket
#.....
import threading

server_name = '127.0.0.1'#IP Adresa e localhostit
server_port = 14000 #NUMRI I PORTIT te serverit
soketiS = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)#Ky rresht krijon soket
in e klientit te quajtur soketiK dhe permban dy parametra.
#Parametri i pare AF_INET-tregon se rrjeti perdore IPv4
#Parametri i dyte SOCK_DGRAM-
tregon se soketi eshte i tipit SOCK DGRAM qe nenkupton qe oshte UDP soket
soketiS.bind((server_name, server_port))#Percaktimi i serverName dhe serverPort p
er lidhje
print('Serveri eshte startuar ne localhost ne portin ' + str(server_port))
print('-----UDP-Serveri-----')
#.....
print('Server eshte duke punuar dhe eshte duke pritur per ndonje kerkese!')
#Funksionet
#.....
#Thread
while True:
    print('-----')
    pergjigjja, address = soketiS.recvfrom(128)#Kerkesa,kontrollimi i bajtave te
mesazhit varesisht kerkeses dhe procesimi i tyre
    print('Klienti eshte lidhur me server %s ne portin %s ' %address)
    print('Kerkesa nga klienti: ' + str(pergjigjja.decode('utf-
8')))#Kerkesa e klientit
    threading._start_new_thread(ThreadUDP,(pergjigjja,address,))#Krijimi i thred
ave
soketiS.close()
```

4. Testimi i programit klient/server ne TCP dhe UDP

TCP Klient/Server



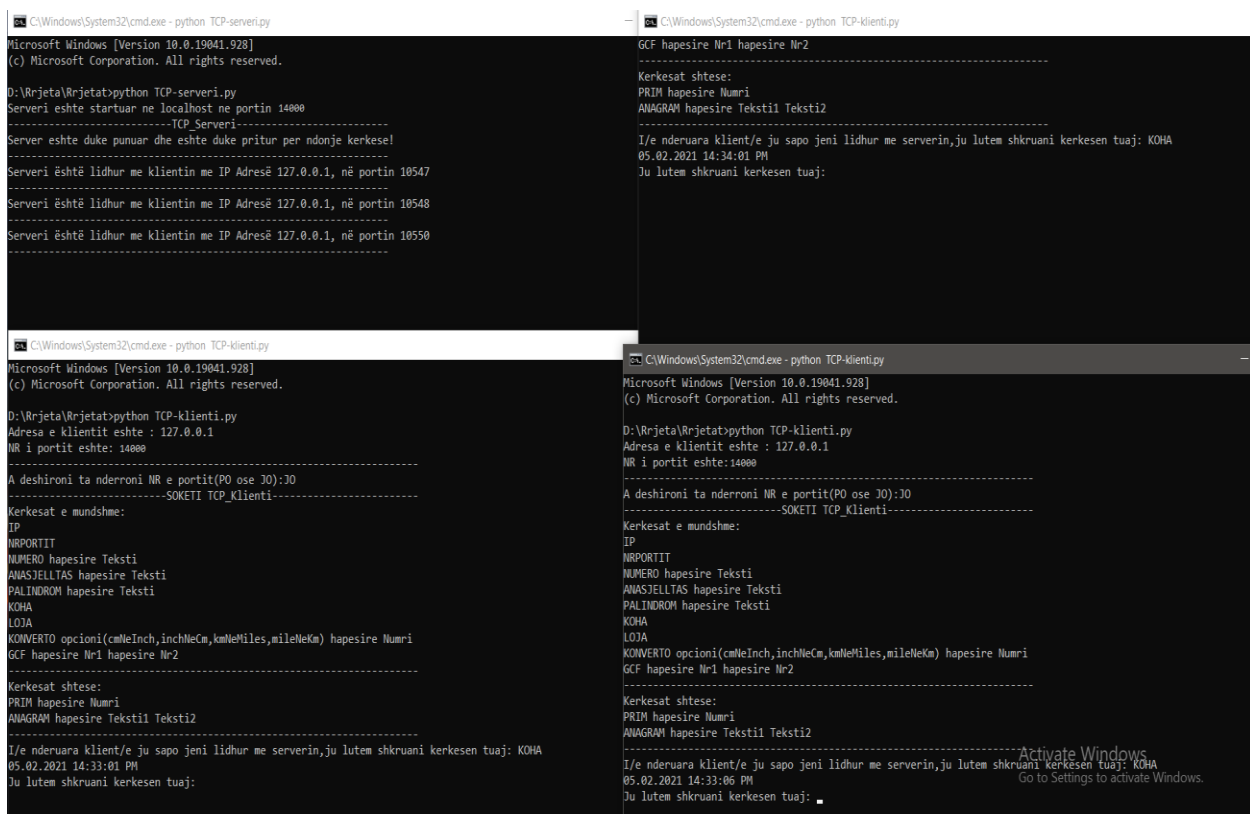
```
1 #Programimi i soketit FIEK Klienti permes protokollit TCP
2
3 import socket #Importi i te gjitha moduleve nga libraria socket
4
5 serverName = '127.0.0.1' #IP Adresa e localhostit
6 serverPort = 14000 #NUMRI I PORTIT te serverit
7 print("Adresa e klientit eshte :",serverName)
8 print("NR i portit eshte:",serverPort)
9 print('-----')
10 pyetje=input("A deshironi ta nderroni NR e portit(PO ose JO):")
11 if(pyetje=='PO' or pyetje=='Po' or pyetje=='po' or pyetje=='pO'):
12     serverName=input("Shkruaj IP adresen:")
13     sP=input("Shkruaj NR e portit:")

LOJA
KONVERTO opcioni(cmNeInch,inchNeCm,kmNeMiles,mileNeKm) hapesire Numri
GCF hapesire Nr1 hapesire Nr2

-----
Kerkesat shtese:
PRIM hapesire Numri
ANAGRAM hapesire Tekstil Teksti2
-----
I/e nderuara klient/e ju sapo jeni lidhur me serverin,ju lutem shkruani kerkesen tuaj: NRPORTIT
Klienti eshte duke perdorur portin 52132
Ju lutem shkruani kerkesen tuaj: 
```

```
Serveri eshte startuar ne localhost ne portin 14000
-----TCP_Serveri-----
Server eshte duke punuar dhe eshte duke pritur per ndonje kerkesa!
Serveri eshte lidhur me klientin me IP Adrese 127.0.0.1, ne portin 52132
-----
```

TCP Klient/Server permes multithread



```
C:\Windows\System32\cmd.exe - python TCP-server.py
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. All rights reserved.

D:\Rrjeta\Rrjetat>python TCP-server.py
Serveri eshte startuar ne localhost ne portin 14000
-----TCP_Serveri-----
Server eshte duke punuar dhe eshte duke pritur per ndonje kerkesa!

Serveri eshte lidhur me klientin me IP Adrese 127.0.0.1, ne portin 10547

Serveri eshte lidhur me klientin me IP Adrese 127.0.0.1, ne portin 10548

Serveri eshte lidhur me klientin me IP Adrese 127.0.0.1, ne portin 10550
-----

C:\Windows\System32\cmd.exe - python TCP-klienti.py
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. All rights reserved.

D:\Rrjeta\Rrjetat>python TCP-klienti.py
Adresa e klientit eshte : 127.0.0.1
NR i portit eshte: 14000

-----SOKETI TCP_Klienti-----
A deshironi ta nderroni NR e portit(PO ose JO):JO

Kerkesat e mundshme:
IP
NRPORTIT
NUMERO hapesire Teksti
ANASJELLTAS hapesire Teksti
PALINDROM hapesire Teksti
KOHA
LOJA
KONVERTO opcioni(cmNeInch,inchNeCm,kmNeMiles,mileNeKm) hapesire Numri
GCF hapesire Nr1 hapesire Nr2

-----
Kerkesat shtese:
PRIM hapesire Numri
ANAGRAM hapesire Tekstil Teksti2
-----
I/e nderuara klient/e ju sapo jeni lidhur me serverin,ju lutem shkruani kerkesen tuaj: KOHA
05.02.2021 14:33:01 PM
Ju lutem shkruani kerkesen tuaj:

C:\Windows\System32\cmd.exe - python TCP-klienti.py
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. All rights reserved.

D:\Rrjeta\Rrjetat>python TCP-klienti.py
Adresa e klientit eshte : 127.0.0.1
NR i portit eshte:14000

-----SOKETI TCP_Klienti-----
A deshironi ta nderroni NR e portit(PO ose JO):JO

Kerkesat e mundshme:
IP
NRPORTIT
NUMERO hapesire Teksti
ANASJELLTAS hapesire Teksti
PALINDROM hapesire Teksti
KOHA
LOJA
KONVERTO opcioni(cmNeInch,inchNeCm,kmNeMiles,mileNeKm) hapesire Numri
GCF hapesire Nr1 hapesire Nr2

-----
Kerkesat shtese:
PRIM hapesire Numri
ANAGRAM hapesire Tekstil Teksti2
-----
I/e nderuara klient/e ju sapo jeni lidhur me serverin,ju lutem shkruani kerkesen tuaj: KOHA
05.02.2021 14:33:06 PM
Ju lutem shkruani kerkesen tuaj: 
```


UDP Klient/Server

```
TCP-klienti.py  UDP-klienti.py X  UDP-serveri.py  TCP-serveri.py

UDP-klienti.py > ...
1  #Programimi i soketit FIEK Klienti permes protokollit TCP
2  import socket #Importi i te gjitha moduleve nga libraria socket
3
4  server_name = 'localhost'#IP Adresa e localhostit
5  server_port = 14000 #NUMRI I PORTIT te serverit
6  print("Adresa e klientit eshte :",server_name)
7  print("NR i portit eshte:",server_port)
8  print('-----')
9  pyetje=input("A deshironi ta nderroni NR e portit(PO ose JO):")
10 if(pyetje=='PO' or pyetje=='Po' or pyetje=='po' or pyetje=='pO'):
11     server_name=input("Shkruaj IP adresen:")
12     sP=input("Shkruaj NR e portit:")
13     server_port=int(sP)

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

-----SOKETI UDP_Klienti-----
Kerkesat e mundshme:
IP
NRPORTIT
NUMERO hapesire Teksti
ANASJELLTAS hapesire Teksti
PALINDROM hapesire Teksti
KOHA
LOJA
KONVERTO opcioni(cmMeInch,inchMeCm,kmMeMiles,mileMeKm) hapesire Numri
GCF hapesire Nr1 hapesire Nr2
-----
Kerkesat shtese:
PRIM hapesire Numri
ANAGRAM hapesire Tekstil Teksti2
-----
I/e nderuara klient/e ju sapo jeni lidhur me serverin,ju lutem shkruani kerkesen tuaj: NRPORTIT
Klienti eshte duke perdorur portin 53443
PS C:\Users\HP\Desktop\University work\Viti2(sem.4)\Rrjetaushtrime\Socket Programming>
```

UDP Klient/Server permes multithreads

```
C:\Windows\System32\cmd.exe - python UDP-serveri.py
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. All rights reserved.

D:\Rrjeta\Rrjetat>python UDP-serveri.py
python: can't open file 'D:\Rrjeta\Rrjetat\UDP-serveri.py': [Errno 2] No such file or directory

D:\Rrjeta\Rrjetat>python UDP-serveri.py
Serveri eshte startuar ne localhost ne portin 14000
-----UDP Serveri-----
Server eshte duke punuar dhe eshte duke pritur per ndonje kerkesel
Klienti eshte lidhur me server 127.0.0.1 ne portin 49998
Kerkesa nga klienti: IP
-----
Klienti eshte lidhur me server 127.0.0.1 ne portin 50000
Kerkesa nga klienti: KOHA
-----
Klienti eshte lidhur me server 127.0.0.1 ne portin 50006
Kerkesa nga klienti: KOHA
-----

C:\Windows\System32\cmd.exe
D:\Rrjeta\Rrjetat>python UDP-klienti.py
Adresa e klientit eshte : localhost
NR i portit eshte: 14000
-----
A deshironi ta nderroni NR e portit(PO ose JO):IP
-----SOKETI UDP_Klienti-----
Kerkesat e mundshme:
IP
NRPORTIT
NUMERO hapesire Teksti
ANASJELLTAS hapesire Teksti
PALINDROM hapesire Teksti
KOHA
LOJA
KONVERTO opcioni(cmMeInch,inchMeCm,kmMeMiles,mileMeKm) hapesire Numri
GCF hapesire Nr1 hapesire Nr2
-----
Kerkesat shtese:
PRIM hapesire Numri
ANAGRAM hapesire Tekstil Teksti2
-----
I/e nderuara klient/e ju sapo jeni lidhur me serverin,ju lutem shkruani kerkesen tuaj: KOHA
05.02.2021 14:37:03 PM

C:\Windows\System32\cmd.exe
KOHA
LOJA
KONVERTO opcioni(cmMeInch,inchMeCm,kmMeMiles,mileMeKm) hapesire Numri
GCF hapesire Nr1 hapesire Nr2
-----
Kerkesat shtese:
PRIM hapesire Numri
ANAGRAM hapesire Tekstil Teksti2
-----
I/e nderuara klient/e ju sapo jeni lidhur me serverin,ju lutem shkruani kerkesen tuaj: KOHA
05.02.2021 14:36:47 PM

D:\Rrjeta\Rrjetat>
```

5. Përfundimi

Realizimi i gjithë projekti ishte mjaftë sfidues gjatë gjithë kohës por njëkohësisht edhe argetues. Ngase për implementimin e një programi të tillë komunikues vetëm ushtrimet në lëndën Rrjetave Kompjuterike nuk kanë mjaftuar . Andaj është dashur të hulumtojmë dhe analizojmë funksione të ndryshme në baze të kërkesave të projektit. Dhe e gjithë kjo ka rezultuar në përforcimin më të madh të koncepteve ndaj arkitektures klient/server, qasjes përmes soketeve, dërgimit-pranimi të kërkesave si dhe procesimin e tyre.

Programi i implementuar më lartë realizon komunikimin mes Klientit dhe Serverit .Të gjitha metodat e lartë përmendura funksionojnë në mënyrën e duhur. Poashtu edhe implementimi tek serveret TCP/UDP për të komunikuar me disa kliente njëkohësisht përmes threads është funksional.

6. Referencat

- Rajkumar Buyya, “Socket Programming”.
- Joseph M. Dibella , “Socket Programming”.
- James F. Kurose, Keith W. Ross, “Computer Networking”.
- David Ascher, “Learning Python”.
- http://en.wikipedia.org/wiki/Berkeley_sockets.
- https://en.wikipedia.org/wiki/User_Datagram_Protocol
- https://en.wikipedia.org/wiki/Transmission_Control_Protocol
- <https://www.tutorialspoint.com/python/index.html>
- <https://www.tutorialspoint.com/socket-programming-in-python>
- <https://www.geeksforgeeks.org/python-programming-language/>
- <https://www.w3schools.com/python/default.asp>
- <https://www.javatpoint.com/python-functions>
- <https://www.learnpython.org/en/Functions>
- https://www.pitt.edu/~naraehan/python3/user_defined_functions.html
- <https://www.programiz.com/python-programming>
- <https://stackoverflow.com/questions/tagged/python>
- <https://www.codementor.io/community/topic/python>
- <https://diveintopython3.net/strings.html>
- <https://www.techbeamers.com/python-tutorial-write-multithreaded-python-server/>
- https://www.tutorialspoint.com/python3/python_multithreading.htm