

# Generator de pachete de rețea

Echipa: Programatori

Coordonator proiect: Cristian Deacu & Horia Calin

Autor: Stan Rareș-Gabriel



2024

Limbaj Utilizat: Python  
Nivel Dificultate: 6/10  
Domeniu: Telecomunicații  
Tehnologii: Scapy

## Cuprins:

- I. Descriere
- II. Resurse
- III. Implementare
- IV. Imbunatatiri
- V. Concluzii

## I.Descriere

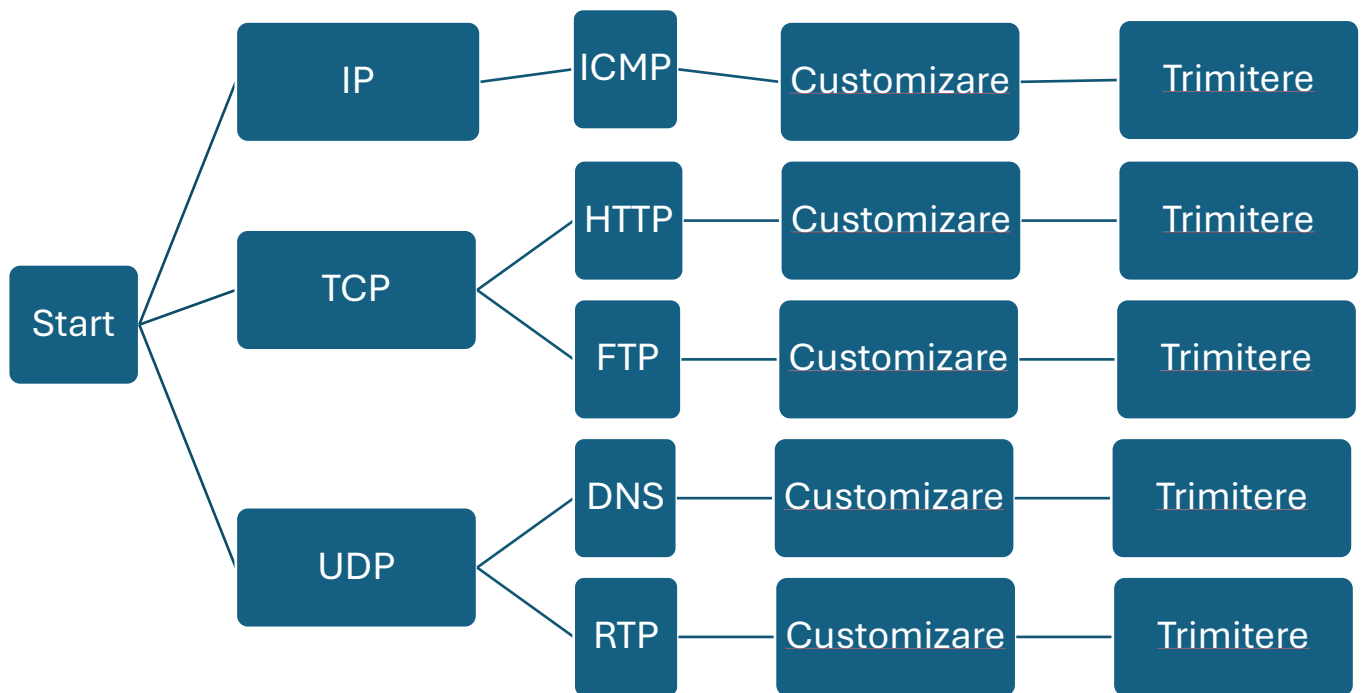
Aplicația este concepută pentru generarea de pachete custom, cât și transmiterea acestora. Pachetele pot fi de 3 tipuri: IP, TCP, UDP și sunt generate cu ajutorul unei interfețe grafice realizată cu biblioteca “tkinter”.

## II. Resurse Folosite

- Limbaj: Python
- Biblioteci:
  - **tkinter** : pentru crearea interfeței grafice
  - **scapy** : pentru manipularea pachetelor de rețea
- Aplicații folosite:
  - **Pycharm**: pentru scrierea și rularea programului
  - **WireShark**: pentru monitorizarea transmiterii pachetelor
- Cursuri utilizate:
  - Documentația oficială Python pentru tkinter
  - Documentația oficială Scapy

### III. Implementare

#### Flowchart:



Aplicațiile de generare și manipulare a pachetelor de rețea sunt esențiale pentru testarea și diagnosticarea rețelelor de comunicații. Un astfel de instrument permite inginerilor de rețea să simuleze diverse scenarii de trafic, să testeze reacția echipamentelor la anumite tipuri de pachete și să identifice potențiale vulnerabilități. În acest context, codul prezentat utilizează Python împreună cu bibliotecile Tkinter și Scapy pentru a crea o interfață grafică de utilizator (GUI) care facilitează crearea și trimiterea de pachete de rețea IP, TCP și UDP. Tkinter oferă instrumentele necesare pentru a dezvolta interfața grafică, iar Scapy este un instrument puternic de manipulare a pachetelor de rețea.

Programul este structurat în mai multe funcții care gestionează diferitele tipuri de pachete de rețea. Interfața principală a aplicației permite utilizatorului să selecteze tipul de pachet pe care dorește să îl trimită (IP, TCP sau UDP), iar pe baza selecției, se deschide o fereastră secundară care colectează detaliile necesare pentru generarea pachetului specific.

Prima parte a codului configurează fereastra principală a aplicației, care conține butoane pentru fiecare tip de pachet (IP, TCP, UDP). La apăsarea fiecărui buton, se deschide o nouă fereastră care solicită informațiile specifice pentru tipul de pachet selectat.

Fereastra principală este creată folosind funcția Tk() din Tkinter și este configurată cu un titlu și un LabelFrame care conține trei butoane pentru selecția tipului de pachet. Acest LabelFrame servește drept container pentru butoane, oferindu-i un titlu și o delimitare clară.

Se definesc mai departe funcții separate pentru fiecare tip de pachet. Aceste funcții creează ferestre secundare unde utilizatorul poate introduce detaliile necesare pentru generarea și trimiterea pachetelor.

Funcția features\_ICMP creează o fereastră pentru pachetele ICMP, solicitând utilizatorului informații precum numărul de pachete, adresa IP de destinație, adresa IP sursă, tipul și codul ICMP. După introducerea informațiilor, utilizatorul poate apăsa butonul "Trimite pachete" pentru a genera și trimite pachetele folosind Scapy.

Utilizatorul introduce detaliile pachetului în câmpurile de text. Aceste detalii includ:

- Numărul de pachete: câmpul în care utilizatorul specifică de câte ori să fie trimis pachetul.
- IP destinație: adresa IP a destinației unde va fi trimis pachetul.
- IP sursă: adresa IP de la care aparent provine pachetul.
- Tip ICMP: tipul mesajului ICMP (de exemplu, tipul 8 pentru "echo request").
- Cod ICMP: codul specific pentru tipul de mesaj ICMP selectat (de obicei 0 pentru "echo request").

La apăsarea butonului "Trimite pachete", se construiește un pachet ICMP folosind Scapy. Pachetul este definit prin concatenarea unui header IP cu un header ICMP.

Pachetul este trimis de numărul de ori specificat de utilizator, folosind funcția `send()` din Scapy. După trimitere, utilizatorul este notificat printr-un mesaj de confirmare.

Procesul de colectare a informațiilor, generare și trimitere a pachetelor este exact la fel pentru toate tipurile de pachete. Principalul lucru diferit pentru fiecare tip de pachet sunt câmpurile ce pot fi modificate.

Câmpurile pentru HTTP (și FTP):

- Numar pachete
- IP destinație
- IP sursă
- Port destinație
- Port sursă
- Flags

Câmpurile pentru DNS (și RTP):

- Numar pachete
- IP destinație
- IP sursă
- Port destinație
- Port sursă
- Payload

Acest proiect demonstrează cum se pot utiliza Tkinter și Scapy pentru a crea o aplicație grafică de generare și trimitere de pachete de rețea. Prin intermediul unei interfețe intuitive, utilizatorii pot introduce detaliile necesare și pot trimite pachete IP, TCP și UDP pentru testarea și diagnosticarea rețelelor. Tkinter oferă un mod simplu de a crea interfețe grafice, în timp ce Scapy permite manipularea detaliată a pachetelor de rețea, oferind o soluție completă pentru gestionarea traficului de rețea în scopuri de testare.



# Exemple de rulare:

## Transmitere pachete ICMP:

Generator de pachete

Tipul de pachete

IP TCP UDP

Selecteaza tipul de pachet IP

ICMP

Numarul de pachete: 100

IP destinatie: 8.8.8.8

IP sursa: 8.8.8.7

Tip ICMP: 8

Cod ICMP: 0

Trimite pachete

Pachete transmise cu succes!

OK

No.	Time	Source	Destination	Protocol	Length	Info
932	73.783447	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
933	73.785293	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
934	73.787232	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
935	73.788848	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
936	73.790762	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
937	73.792380	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
938	73.794241	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
939	73.795826	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
940	73.797935	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
941	73.799777	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
942	73.801600	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
943	73.803937	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
944	73.805788	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
945	73.807515	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
946	73.809215	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
947	73.811024	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
948	73.812833	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
949	73.814389	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
950	73.816075	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
951	73.817786	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
952	73.819607	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
953	73.821085	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
954	73.823019	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
955	73.824459	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
956	73.826435	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
957	73.828195	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
958	73.832791	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
959	73.834440	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
960	73.836348	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
961	73.837957	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
962	73.839544	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
963	73.841179	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
964	73.842947	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)
965	73.844964	8.8.8.7	8.8.8.8	ICMP	42	Echo (ping)

## Transmitere pachete TCP(HTTP):

Generator de pachete

Tipul de pachete

IP TCP UDP

Selecteaza tipul de pachet TCP

HTTP

FTP

HTTP

Numarul de pachete: 100

IP destinatie: 8.8.8.8

IP sursa: 8.8.8.7

Port destinatie: 80

Port sursa: 80

Flags: S

Trimite pachete

Pachete transmise cu succes!

OK

No.	Time	Source	Destination	Protocol	Length	Info
402	31.597481	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
403	31.598710	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
404	31.600152	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
405	31.601372	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
406	31.602875	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
407	31.604322	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
408	31.605735	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
409	31.607194	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
410	31.608826	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
411	31.610082	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
412	31.611823	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
413	31.613469	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
414	31.614763	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
415	31.616245	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
416	31.617684	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
417	31.619285	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
418	31.620730	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
419	31.622221	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
420	31.623781	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
421	31.625388	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
422	31.628368	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
423	31.630503	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
424	31.632006	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
425	31.633440	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
426	31.634984	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
427	31.636318	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
428	31.637833	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
429	31.639108	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
430	31.640540	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
431	31.642050	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
432	31.643324	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
433	31.645347	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]
434	31.646922	8.8.8.7	8.8.8.8	TCP	54	[TCP Retransmission]

## Transmitere pachete UDP(DNS):

The screenshot shows a network packet capture interface with three main windows:

- Generator de pachete**: A window for generating packets, with buttons for IP, TCP, and UDP.
- Selecteaza tipul de pachet UDP**: A window to select the type of UDP packet, with buttons for DNS and RTP.
- DNS**: A window showing details for a selected DNS packet, including:
  - Numarul de pachete: 100
  - Destinatie: 8.8.8.8
  - Sursa: 8.8.8.7
  - Port destinatie: 53
  - Port sursa: 53
  - Payload: Hello, UDP!
  - Trimitere pachete button

The main window displays a list of captured packets with the following columns: No., Time, Source, Destination, Protocol, Length, and Info.

No.	Time	Source	Destination	Protocol	Length	Info
403	40.591853	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
404	40.593234	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
405	40.594807	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
406	40.596293	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
407	40.597887	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
408	40.599191	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
409	40.600975	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
410	40.602540	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
411	40.603834	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
412	40.605469	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
413	40.606774	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
414	40.608215	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
415	40.609520	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
416	40.611104	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
417	40.612435	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
418	40.614036	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
419	40.615657	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
420	40.617117	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
421	40.618950	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
422	40.620476	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
423	40.622009	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
424	40.623468	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
425	40.624998	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
426	40.626617	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
427	40.628495	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
428	40.629994	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
429	40.631614	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
430	40.633082	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
431	40.634361	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
432	40.635931	8.8.8.7	8.8.8.8	DNS	53	Unknown operation
433	40.637217	8.8.8.7	8.8.8.8	DNS	53	Unknown operation

## IV. Imbunatatiri

- Capturarea pachetelor poate fi limitată de configurațiile firewall-ului sau ale rețelei.
- Adăugarea opțiunilor pentru configurarea avansată a pachetelor.
- Implementarea unui mecanism de logare a activității.
- Adăugarea mai multor tipuri de pachete
- Adăugarea unor mesaje de eroare la scrierea greșită a datelor de către utilizator

## V. Concluzii

- Desi este un program destul de lung, majoritatea comenzilor sunt doar pentru crearea interfeței grafice
- Este folositoare pentru diagnosticarea rețelei și testarea comunicării între calculatoare

## VI. Anexa

```
import tkinter
from tkinter.ttk import Combobox
from scapy.all import *
from tkinter import ttk
from tkinter import messagebox

def features_ICMP():
    rootICMP = tkinter.Tk()
    rootICMP.title("ICMP")
    rootICMP.geometry("300x200")
    rootICMP.resizable(False, False)
    label_nr = ttk.Label(rootICMP, text="Numarul de pachete:")
    label_nr.grid(row=0, column=0)
    entry_nr = ttk.Entry(rootICMP, width=30)
    entry_nr.grid(row=0, column=1)
    entry_nr.insert(0, "10")
    label_d = ttk.Label(rootICMP, text="IP destinatie:")
    label_d.grid(row=1, column=0)
    entry_d = ttk.Entry(rootICMP, width=30)
    entry_d.grid(row=1, column=1)
    entry_d.insert(0, "127.0.0.1")
    label_s = ttk.Label(rootICMP, text="IP sursa:")
    label_s.grid(row=2, column=0)
    entry_s = ttk.Entry(rootICMP, width=30)
    entry_s.grid(row=2, column=1)
    entry_s.insert(0, "127.0.0.1")
    label_tip = ttk.Label(rootICMP, text="Tip ICMP:")
    label_tip.grid(row=3, column=0)
    entry_tip = ttk.Entry(rootICMP, width=30)
    entry_tip.grid(row=3, column=1)
    entry_tip.insert(0, "8")
    label_cod = ttk.Label(rootICMP, text="Cod ICMP:")
    label_cod.grid(row=4, column=0)
    entry_cod = ttk.Entry(rootICMP, width=30)
    entry_cod.grid(row=4, column=1)
    entry_cod.insert(0, "0")
    def trimiteICMP():
        icmp_packet = IP(src=entry_s.get(), dst=entry_d.get()) /
        ICMP(type=int(entry_tip.get()), code=int(entry_cod.get()))
        print(icmp_packet.show())
        for i in range(int(entry_nr.get())):
            send(icmp_packet)
        messagebox.showinfo("!!!", "Pachete transmise cu succes!")
    buttonICMP = ttk.Button(rootICMP, text="Trimite pachete", command=
    trimiteICMP)
    buttonICMP.grid(row=6, column=0)
def features_HTTP():
    rootHTTP = tkinter.Tk()
    rootHTTP.title("HTTP")
    rootHTTP.geometry("300x150")
    rootHTTP.resizable(False, False)
    label_nr = ttk.Label(rootHTTP, text="Numarul de pachete:")
    label_nr.grid(row=0, column=0)
    entry_nr = ttk.Entry(rootHTTP, width=30)
    entry_nr.grid(row=0, column=1)
    entry_nr.insert(0, "1")
```

```

label_d = ttk.Label(rootHTTP, text="IP destinatie:")
label_d.grid(row=1, column=0)
entry_d = ttk.Entry(rootHTTP, width=30)
entry_d.grid(row=1, column=1)
entry_d.insert(0, "127.0.0.1")
label_s = ttk.Label(rootHTTP, text="IP dursa:")
label_s.grid(row=2, column=0)
entry_s = ttk.Entry(rootHTTP, width=30)
entry_s.grid(row=2, column=1)
entry_s.insert(0, "127.0.0.1")
label_pd = ttk.Label(rootHTTP, text="Port destinatie:")
label_pd.grid(row=3, column=0)
entry_pd = ttk.Entry(rootHTTP, width=30)
entry_pd.grid(row=3, column=1)
entry_pd.insert(0, "80")
label_ps = ttk.Label(rootHTTP, text="Port sursa:")
label_ps.grid(row=4, column=0)
entry_ps = ttk.Entry(rootHTTP, width=30)
entry_ps.grid(row=4, column=1)
entry_ps.insert(0, "80")
def trimiteTCP_HTTP():
    tcp_http_packet = IP(src=entry_s.get(), dst=entry_d.get()) /
TCP(sport=int(entry_ps.get()), dport=int(entry_pd.get()))
    for i in range(int(entry_nr.get())):
        send(tcp_http_packet)
        messagebox.showinfo("!!!", "Pachete transmise cu succes!")
buttonTCP_HTTP = ttk.Button(rootHTTP, text="Trimite pachete",
command=trimiteTCP_HTTP)
buttonTCP_HTTP.grid(row=5, column=0)
def features_FTP():
    rootFTP = tkinter.Tk()
    rootFTP.title("FTP")
    rootFTP.geometry("300x150")
    rootFTP.resizable(False, False)
    label_nr = ttk.Label(rootFTP, text="Numarul de pachete:")
    label_nr.grid(row=0, column=0)
    entry_nr = ttk.Entry(rootFTP, width=30)
    entry_nr.grid(row=0, column=1)
    entry_nr.insert(0, "1")
    label_d = ttk.Label(rootFTP, text="Destinatie:")
    label_d.grid(row=1, column=0)
    entry_d = ttk.Entry(rootFTP, width=30)
    entry_d.grid(row=1, column=1)
    entry_d.insert(0, "127.0.0.1")
    label_s = ttk.Label(rootFTP, text="Sursa:")
    label_s.grid(row=2, column=0)
    entry_s = ttk.Entry(rootFTP, width=30)
    entry_s.grid(row=2, column=1)
    entry_s.insert(0, "127.0.0.1")
    label_pd = ttk.Label(rootFTP, text="Port destinatie:")
    label_pd.grid(row=3, column=0)
    entry_pd = ttk.Entry(rootFTP, width=30)
    entry_pd.grid(row=3, column=1)
    entry_pd.insert(0, "20")
    label_ps = ttk.Label(rootFTP, text="Port sursa:")
    label_ps.grid(row=4, column=0)
    entry_ps = ttk.Entry(rootFTP, width=30)
    entry_ps.grid(row=4, column=1)
    entry_ps.insert(0, "21")
    def trimiteTCP_FTP():
        tcp ftp packet = IP(src=entry_s.get(), dst=entry_d.get()) /

```

```

TCP(sport=int(entry_ps.get()), dport=int(entry_pd.get()))
    for i in range(int(entry_nr.get())):
        send(tcp_ftp_packet)
        messagebox.showinfo("!!!", "Pachete transmise cu succes!")
    buttonTCP_FTP = ttk.Button(rootFTP, text="Trimite pachete",
command=trimiteTCP_FTP)
    buttonTCP_FTP.grid(row=5, column=0)
def features_DNS():
    rootDNS = tkinter.Tk()
    rootDNS.title("DNS")
    rootDNS.geometry("300x180")
    rootDNS.resizable(False, False)
    label_nr = ttk.Label(rootDNS, text="Numarul de pachete:")
    label_nr.grid(row=0, column=0)
    entry_nr = ttk.Entry(rootDNS, width=30)
    entry_nr.grid(row=0, column=1)
    entry_nr.insert(0, "1")
    label_d = ttk.Label(rootDNS, text="Destinatie:")
    label_d.grid(row=1, column=0)
    entry_d = ttk.Entry(rootDNS, width=30)
    entry_d.grid(row=1, column=1)
    entry_d.insert(0, "127.0.0.1")
    label_s = ttk.Label(rootDNS, text="Sursa:")
    label_s.grid(row=2, column=0)
    entry_s = ttk.Entry(rootDNS, width=30)
    entry_s.grid(row=2, column=1)
    entry_s.insert(0, "127.0.0.1")
    label_pd = ttk.Label(rootDNS, text="Port destinatie:")
    label_pd.grid(row=3, column=0)
    entry_pd = ttk.Entry(rootDNS, width=30)
    entry_pd.grid(row=3, column=1)
    entry_pd.insert(0, "53")
    label_ps = ttk.Label(rootDNS, text="Port sursa:")
    label_ps.grid(row=4, column=0)
    entry_ps = ttk.Entry(rootDNS, width=30)
    entry_ps.grid(row=4, column=1)
    entry_ps.insert(0, "53")
    label_pl = ttk.Label(rootDNS, text="Payload:")
    label_pl.grid(row=5, column=0)
    entry_pl = ttk.Entry(rootDNS, width=30)
    entry_pl.grid(row=5, column=1)
    entry_pl.insert(0, "Hello, UDP!")
    def trimiteUDP_DNS():
        payload = entry_pl.get().encode()
        udp_dns_packet = IP(src=entry_s.get(), dst=entry_d.get()) /
UDP(sport=int(entry_ps.get()), dport=int(entry_pd.get())) / payload
        for i in range(int(entry_nr.get())):
            send(udp_dns_packet)
            messagebox.showinfo("!!!", "Pachete transmise cu succes!")
        buttonUDP_DNS = ttk.Button(rootDNS, text="Trimite pachete",
command=trimiteUDP_DNS)
        buttonUDP_DNS.grid(row=6, column=0)
def features_RTP():
    rootRTP = tkinter.Tk()
    rootRTP.title("RTP")
    rootRTP.geometry("300x180")
    rootRTP.resizable(False, False)
    label_nr = ttk.Label(rootRTP, text="Numarul de pachete:")
    label_nr.grid(row=0, column=0)
    entry_nr = ttk.Entry(rootRTP, width=30)
    entry_nr.grid(row=0, column=1)

```

```

entry_nr.insert(0, "1")
label_d = ttk.Label(rootRTP, text="Destinatie:")
label_d.grid(row=1, column=0)
entry_d = ttk.Entry(rootRTP, width=30)
entry_d.grid(row=1, column=1)
entry_d.insert(0, "127.0.0.1")
label_s = ttk.Label(rootRTP, text="Sursa:")
label_s.grid(row=2, column=0)
entry_s = ttk.Entry(rootRTP, width=30)
entry_s.grid(row=2, column=1)
entry_s.insert(0, "127.0.0.1")
label_pd = ttk.Label(rootRTP, text="Port destinatie:")
label_pd.grid(row=3, column=0)
entry_pd = ttk.Entry(rootRTP, width=30)
entry_pd.grid(row=3, column=1)
entry_pd.insert(0, "5004")
label_ps = ttk.Label(rootRTP, text="Port sursa:")
label_ps.grid(row=4, column=0)
entry_ps = ttk.Entry(rootRTP, width=30)
entry_ps.grid(row=4, column=1)
entry_ps.insert(0, "5004")
label_pl = ttk.Label(rootRTP, text="Payload:")
label_pl.grid(row=5, column=0)
entry_pl = ttk.Entry(rootRTP, width=30)
entry_pl.grid(row=5, column=1)
entry_pl.insert(0, "Hello, UDP!")
def trimiteUDP_RTP():
    payload = entry_pl.get().encode()
    udp_dns_packet = IP(src=entry_s.get(), dst=entry_d.get()) /
UDP(sport=int(entry_ps.get()), dport=int(entry_pd.get())) / payload
    for i in range(int(entry_nr.get())):
        send(udp_dns_packet)
        messagebox.showinfo("!!!", "Pachete transmise cu succes!")
    buttonUDP_RTP = ttk.Button(rootRTP, text="Trimite pachete",
command=trimiteUDP_RTP)
    buttonUDP_RTP.grid(row=6, column=0)
def new_IP():
    rootIP = tkinter.Tk()
    rootIP.title("Selecteaza tipul de pachet IP")
    rootIP.geometry("350x100")
    rootIP.resizable(False, False)
    rootIP.columnconfigure(0, weight=1)
    rootIP.rowconfigure(0, weight=1)
    buttonIP = ttk.Button(rootIP, text="ICMP", command=features_ICMP)
    buttonIP.grid(row=0, column=0)

def new_TCP():
    rootTCP = tkinter.Tk()
    rootTCP.title("Selecteaza tipul de pachet TCP")
    rootTCP.geometry("350x100")
    rootTCP.resizable(False, False)
    rootTCP.columnconfigure(0, weight=1)
    rootTCP.rowconfigure(0, weight=1)
    buttonTCP_HTTP = ttk.Button(rootTCP, text="HTTP",
command=features_HTTP)
    buttonTCP_HTTP.grid(row=0, column=0)
    buttonTCP_FTP = ttk.Button(rootTCP, text="FTP", command=features_FTP)
    buttonTCP_FTP.grid(row=1, column=0)

def new_UDP():

```



```
rootUDP = tkinter.Tk()
rootUDP.title("Selecteaza tipul de pachet UDP")
rootUDP.geometry("350x100")
rootUDP.resizable(False, False)
rootUDP.columnconfigure(0, weight=1)
rootUDP.rowconfigure(0, weight=1)
buttonUDP_DNS = ttk.Button(rootUDP, text="DNS", command=features_DNS)
buttonUDP_DNS.grid(row=0, column=0)
buttonUDP_RTP = ttk.Button(rootUDP, text="RTP", command=features_RTP)
buttonUDP_RTP.grid(row=1, column=0)

root = tkinter.Tk()
root.title("Generator de pachete")
packet_type_frame = ttk.LabelFrame(root, text="Tipul de pachete")
packet_type_frame.grid(row=0, column=0, padx=10, pady=10, sticky="ew")
buttonIP = ttk.Button(packet_type_frame, text="IP", command=new_IP)
buttonIP.grid(row=1, column=0, padx=10, pady=10)
buttonTCP = ttk.Button(packet_type_frame, text="TCP", command=new_TCP)
buttonTCP.grid(row=1, column=1, padx=10, pady=10)
buttonUDP = ttk.Button(packet_type_frame, text="UDP", command=new_UDP)
buttonUDP.grid(row=1, column=2, padx=10, pady=10)

root.mainloop()
```