

Sieci Bezprzewodowe

---

## Sieci Bezprzewodowe - laboratorium 1/2

---

Politechnika Śląska  
Wydział Matematyki Stosowanej

Bartosz Jarzyński  
Jakub Chrobok



**Politechnika  
Śląska**

Gliwice, 8 marca 2023

# 1 Opis Działania Programu

Założeniem zadania jest nawiązanie połączenia między dwoma podmiotami - serwerem oraz klientem. Program zaczytuje od użytkownika adresy IP i porty dwóch klientów, następnie rozpoczyna nasłuchiwanie.

Oprogramowanie umożliwia jednocześnie odbieranie i nadawanie danych.

Foldery klienckie zawierają ten sam kod. Programy zaczytują adres IP, port serwera i port kliencki. Rozmiar bufora został ustalony na 1024. Następnie oprogramowanie tworzy Socket UDP dla strony klienta oraz binduje go do sztywno dla ustalonego adresu IP.

## 2 Kod źródłowy

```
import threading
import socket

srvIP = input("Podaj serwer IP: ")
srvPort = int(input("Podaj port serwera: "))
cntPort = int(input("Podaj port klienta:"))
srvAddrPort = (srvIP, srvPort)
buff = 1024

UDPCnt = socket.socket(family=socket.AF_INET, type = socket.SOCK_DGRAM)
UDPCnt.bind(("127.0.0.1", cntPort))
UDPCnt.sendto("Połączono.".encode(), srvAddrPort)

def rcv():
    while True:
        try:
            wd, _ = UDPCnt.recvfrom(buff)
            print(wd.decode())
        except:
            pass
thread = threading.Thread(target=rcv)
thread.start()

while True:
    wd = input("")
    if wd == "$quit":
        exit()
    else:
        UDPCnt.sendto(wd.encode(), srvAddrPort)
```

Obraz 1: Kod klienta.

```
import threading
import socket
import queue, string, random

hostLokalny = "127.0.0.1"
portLokalny = 5555
buff = 1024

UDPSrv = socket.socket(family=socket.AF_INET, type = socket.SOCK_DGRAM)
UDPSrv.bind((hostLokalny, portLokalny))

wdmnc = queue.Queue()
klienci = []

cntIP_1 = input("Podaj IP pierwszego klienta:")
cntPort_1 = int(input("Podaj port pierwszego klienta:"))
cntIP_2 = input("Podaj IP drugiego klienta:")
cntPort_2 = int(input("Podaj port drugiego klienta:"))

czZakl = input("Podaj częstotliwość zakłócenia - 0 lub 1: ")
print("Serwer UDP nasłuchuje.")

def dsrptr(wdmnc, czZakl):
    err = ""
    czZakl = float(czZakl)
    chr = string.ascii_letters + string.digits + string.punctuation
    for letter in err:
        rndnum = random.uniform(0,1)
        if czZakl >= rndnum:
            letter = random.choice(chr)
            err += letter
        else:
            err += letter
    return err
```

Obraz 2: Kod serwera: część 1.

```
def rcv():
    while True:
        try:
            wd, addr = UDPSrv.recvfrom(buff)
            if (addr[0] == cntIP_1 and addr[1] == cntPort_1) or (addr[0] == cntIP_2 and addr[1] == cntPort_2):
                wdmisc.put((wd,addr))
            else: pass
        except: pass

def strumien():
    while True:
        while not wdmisc.empty():
            wd, addr = wdmisc.get()
            print(addr)

            wd = wd.decode()
            print("Pierwotny klient " + addr[0] + ": " + wd)

            wd = dsrptry(wd, czZakl)
            print("Klient " + addr[0] + ": " + wd)
            wd = wd.encode()

            if addr not in klienci:
                klienci.append(addr)
                for klient in klienci:
                    UDPSrv.sendto((addr[0] + " połączony.").encode(), klient)
            for klient in klienci:
                UDPSrv.sendto(wd, klient)

thread1 = threading.Thread(target=rcv)
thread2 = threading.Thread(target=strumien)

thread1.start()
thread2.start()
```

Obraz 3: Kod serwera: część 2.