# Bansilal Ramnath Agarwal Charitable Trust’s

Vishwakarma Institute of Technology,Pune-37

*(An Autonomous Institute of Savitribai Phule Pune University)*

**Assignment 10:**

 **Department of Artificial Intelligence and Data Science**

|  |  |
| --- | --- |
| **Division** | A |
| **Batch** | 1 |
| **Rollno** | 23 |
| **Name** | Avishkar Ghodke |

**CODE :**

**Sender.py**import socket

import os

import time

# 🔹 Receiver IP and Port (Change as per your network)

SERVER\_IP = "192.168.1.100"  # Change to the receiver's IP

SERVER\_PORT = 5005

# 🔹 File to Send (Update the path)

FILE\_TO\_SEND = r"C:\Users\jines\OneDrive\Desktop\SY SEM 2\CN\sample.txt"  # Change this path

# 🔹 UDP Config

CHUNK\_SIZE = 1024  # 1KB chunk size

def send\_file(filename):

    """Send a file to the receiver using UDP."""

    if not os.path.isfile(filename):

        print(f"Error: File '{filename}' not found!")

        return

    sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

    # Get file size

    file\_size = os.path.getsize(filename)

    # Send filename and size

    sock.sendto(f"{os.path.basename(filename)}|{file\_size}".encode(), (SERVER\_IP, SERVER\_PORT))

    # Open file and send in chunks

    with open(filename, "rb") as f:

        bytes\_sent = 0

        while True:

            chunk = f.read(CHUNK\_SIZE)

            if not chunk:

                break  # EOF

            sock.sendto(chunk, (SERVER\_IP, SERVER\_PORT))

            bytes\_sent += len(chunk)

            # Wait for ACK

            try:

                sock.settimeout(0.5)  # 500ms timeout

                ack, \_ = sock.recvfrom(1024)

                if ack.decode() != "ACK":

                    print("ACK not received, resending last chunk")

                    bytes\_sent -= len(chunk)

                    f.seek(bytes\_sent)

            except socket.timeout:

                print("Timeout! Resending last chunk")

                bytes\_sent -= len(chunk)

                f.seek(bytes\_sent)

    print(f"✅ File '{filename}' sent successfully.")

    sock.close()

if \_\_name\_\_ == "\_\_main\_\_":

send\_file(FILE\_TO\_SEND)



**Receiver.py**

import socket

import os

# 🔹 Receiver IP and Port (Listen for incoming files)

RECEIVER\_IP = "0.0.0.0"  # Listen on all network interfaces

RECEIVER\_PORT = 5005

# 🔹 UDP Config

CHUNK\_SIZE = 1024  # 1KB chunk size

def receive\_file():

    """Receive a file over UDP and save it."""

    sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

    sock.bind((RECEIVER\_IP, RECEIVER\_PORT))

    print(f"📥 Listening for incoming files on {RECEIVER\_IP}:{RECEIVER\_PORT}...")

    # Receive file metadata

    data, sender\_addr = sock.recvfrom(1024)

    filename, file\_size = data.decode().split("|")

    file\_size = int(file\_size)

    # Create the received file

    save\_path = os.path.join(os.getcwd(), filename)

    with open(save\_path, "wb") as f:

        bytes\_received = 0

        while bytes\_received < file\_size:

            chunk, sender\_addr = sock.recvfrom(CHUNK\_SIZE)

            f.write(chunk)

            bytes\_received += len(chunk)

            # Send ACK to sender

            sock.sendto("ACK".encode(), sender\_addr)

    print(f"✅ File '{filename}' received successfully and saved as '{save\_path}'.")

    sock.close()

if \_\_name\_\_ == "\_\_main\_\_":

    receive\_file()

:

