```python
""" Sender program for Cosc264 Assignment

    Authors: Josh Bernasconi 68613585
             James Toohey    27073776
"""

import socket
import select
import sys
from helpers import *
from packet import Packet
import time


def sender(Sin_port, Sout_port, CSin_port, filename):
    """ Checks ports, sets up connections, then hands over to the main loop """

    ports_ok = check_ports(Sin_port, Sout_port, CSin_port)

    if ports_ok:
        print("Port numbers all valid\n")
    else:
        print("There is a problem with the supplied port numbers!\n Exiting")
        sys.exit()

    file = open(filename, "rb")  # Check it exists. If not, exit.

    # Socket init
    Sin = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    Sout = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    CSin = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    Sin.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    Sout.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    CSin.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

    # Bind
    try:
        print("Binding port Rin")
        Sin.bind(('localhost', Sin_port))
        print("Rin successfully bound\n")
        print("Binding port Rout")
        Sout.bind(('localhost', Sout_port))
        print("Rout successfully bound\n")
    except socket.error as msg:
        print("Bind failed. Exiting.\n Error: " + str(msg))
        sys.exit()

    # Try to connect Sout 5 times before giving up (waiting 5 seconds between attempts)
    connected = False
    connect_attempts = 0
    while not connected:
        try:
            print("Connecting Sout to CSin")
            Sout.connect(('localhost', CSin_port))
            print("Connection successful\n")
            connected = True
        except socket.error as msg:
            connect_attempts += 1
            if msg.errno in [111, 10061] and connect_attempts < 6:
                print("Connection refused {} time(s), sleeping and retrying".format(connect_attempts))
                time.sleep(5)
                pass
            else:
                print("Connect failed. Exiting\n Error: " + str(msg))
                sys.exit()

    # Listen and Accept Sin
    Sin.listen(50)
    Sin, _ = Sin.accept()

    # Read file
    next, size, count = read_file(Sin, Sout, file)

    last_packet = Packet(0, next, 0, "")
    data_packet = pack_data(last_packet)
    Sout.send(data_packet)
    print("Sent {} bytes".format(size))
    print("Number needed for perfect transmission: {}".format(size//512))
```

```
    print("Actually took: {}".format(count))

    Sin.close()
    Sout.close()
    CSin.close()

def read_file(Sin, Sout, file):
    """
    An outer loop which reads the file and sends packets of its content. Also receives
    acknowledgement packets to ensure successful delivery of packets.
    """
    byte_file = file.read()
    n = len(byte_file)
    size = n
    sent = 0
    count = 0
    next = 0
    exit_flag = False

    # Send
    while not exit_flag:
        if n == 0:
            packet = Packet(0, next, 0, '')
            data_packet = pack_data(packet)
            exit_flag = True
        else:
            if n - sent > 512:
                data = byte_file[sent:sent + 512]
                packet = Packet(0, next, 512, data)
                data_packet = pack_data(packet)
                sent += 512
            else:
                data = byte_file[sent:]
                packet = Packet(0, next, len(data), data)
                data_packet = pack_data(packet)
                print("Last data packet sent")
                exit_flag = True

        count, next, exit_flag = check(Sin, Sout, count, data_packet, next, file, exit_flag)

    return next, size, count


def check(Sin, Sout, count, data_packet, next, file, exit_flag):
    """An inner loop which checks that the packet has been successfully sent."""
    successfully_sent = False
    while not successfully_sent:
        count += 1
        Sout.send(data_packet)
        readable, _, _ = select.select([Sin], [], [], 1)  # Timeout after 1 second. If timeout, retransmit.

        if len(readable) == 1:
            data_in, address = readable[0].recvfrom(1024)
            rcvd, valid_packet = get_packet(data_in)
            if valid_packet and rcvd.pac_type == 1 and rcvd.data_len == 0 and rcvd.seqno == next:
                next = 1 - next
                print("Received acknowledgement packet.")
                successfully_sent = True
                if exit_flag:  # Go to beginning of outer loop
                    file.close()
                    break
        else:  # retransmit
            print("timed out")
            print("Resending packet")
    return count, next, exit_flag


if __name__ == '__main__':

    if len(sys.argv) != 5:
        print("Invalid command.")
        print("Usage: sender.py [Sin port] [Sout port] [CSin port] input_filename")
    else:
        Sin = int(sys.argv[1])
        Sout = int(sys.argv[2])
        CSin = int(sys.argv[3])
        filename = sys.argv[4]

        sender(Sin, Sout, CSin, filename)
```