```python
""" Receiver program for Cosc264 Assignment

    Authors: Josh Bernasconi 68613585
             James Toohey    27073776
"""

import socket
import sys
import select
import os.path
import time

from helpers import *
from packet import Packet


def receiver(Rin_port, Rout_port, CRin_port, filename):
    """ Checks ports, sets up connections, then hands over to the main loop """

    ports_ok = check_ports(Rin_port, Rout_port, CRin_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()

    if not os.path.isfile(filename):
        file = open(filename, "wb+")
    else:
        print("File already exists, aborting")
        sys.exit()

    # Socket init
    Rin = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    Rout = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    CRin = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    Rin.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    Rout.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    CRin.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

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

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

    # Connect Rout to CRin
    connected = False
    connect_attempts = 0
    while not connected:
        try:
            print("Connecting Rout to CRin")
            Rout.connect(('localhost', CRin_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()
```

```
    # Read/Write
    read_and_write(Rin, Rout, file)

    Rin.close()
    Rout.close()
    CRin.close()

    return None


def read_and_write(Rin, Rout, file):
    """ Receiver the packets, check validity, acknowledge, then write to file if valid """
    expected = 0
    finished = False
    while not finished:  # while the empty packet has not been found
        readable, _, _ = select.select([Rin], [], [])
        if len(readable) == 1:
            data_in, address = readable[0].recvfrom(1024)
            # print(len(data_in))
            # if len(data_in) == 0:
            #     print("Finished, I think...")
            #     finished = True
            #     continue
            rcvd, valid_packet = get_packet(data_in)
            if not valid_packet:
                print("Invalid packet, stop processing\n")
                continue
            elif rcvd.pac_type == 1:
                print("Packet type not dataPacket, stop processing\n")
                continue
            elif rcvd.seqno != expected:
                acknowledge(Rout, rcvd.seqno, expected)
                print("out of sequence")
                continue

            if rcvd.data_len > 0:
                print("Received valid data packet, writing...")
                acknowledge(Rout, rcvd.seqno, expected)
                file.write(rcvd.data)
                expected = 1 - expected
            else:
                print("Finished")
                finished = True


def acknowledge(Rout, seqno, expected):
    """Creates appropriate acknowledgement packets and sends them through Rout."""
    if seqno != expected:
        packet = Packet(1, seqno, 0, "")  # Still needs a data parameter. Page 6
        acknowledgement_packet = pack_data(packet)
        Rout.send(acknowledgement_packet)

    elif seqno == expected:
        packet = Packet(1, seqno, 0, "")
        acknowledgement_packet = pack_data(packet)
        Rout.send(acknowledgement_packet)


if __name__ == '__main__':
    if len(sys.argv) != 5:
        print("Invalid command.")
        print("Usage: receiver.py [Rin port] [Rout port] [CRin port] dest_filename")
    else:
        Rin = int(sys.argv[1])
        Rout = int(sys.argv[2])
        CRin = int(sys.argv[3])
        filename = sys.argv[4]

        receiver(Rin, Rout, CRin, filename)
```