```python
""" extra functions and classes used by channel, receiver and sender

    Authors: Josh Bernasconi 68613585
             James Toohey    27073776
"""

from struct import *
from packet import Packet
from packet import Header


def check_ports(*ports):
    """ Returns True if there are no duplicate ports and if all
        ports are within the acceptable range, False otherwise.
        """

    all_clear = True
    set_ports = set(ports)

    if len(ports) != len(set_ports):  # Check for duplicate ports
        all_clear = False

    for port in ports:  # check each port is within acceptable port range
        if port < 1024 or port > 64000 or not (isinstance(port, int)):
            all_clear = False

    return all_clear


def pack_data(packet):
    if type(packet.data) != bytes:
        packed = pack('!2I3i' + str(packet.data_len) + 's',
                      packet.magicno, packet.checksum, packet.pac_type, packet.seqno, packet.data_len,
                      bytes(packet.data, 'utf8'))
    else:
        packed = pack('!2I3i' + str(packet.data_len) + 's',
                      packet.magicno, packet.checksum, packet.pac_type, packet.seqno, packet.data_len, packet.data)

    return packed


def get_header(packet):
    header = unpack('!2I3i', packet[:20])

    return header


def get_header_object(packet):
    header_object = Header(get_header(packet))

    return header_object


def get_data(packet, data_len):
    data = unpack(str(data_len) + 's', packet[20:20 + data_len])
    return data[0]


def get_packet(in_data):
    valid_packet = False
    packet = None

    if in_data != b'':
        header = get_header(in_data)
        checksum = header[1]
        pac_type = header[2]
        seq_no = header[3]
        data_len = header[4]

        if data_len >= len(in_data) - 20:
            data = get_data(in_data, data_len)

            packet = Packet(pac_type, seq_no, data_len, data, checksum)

            valid_packet = packet.checksum == packet.calculate_checksum()

    return packet, valid_packet
```