```python
"""
COSC364 2022-S1 Assignment: RIP routing
Authors: MENG ZHANG (71682325), ZHENG CHAO (21671773)
File: router_interface.py
"""

# Import Modules
import socket
import select


# Router Network Interface Class
class Interface:
    """
    A router interface object which includes:
    * Multiple sockets with corresponding ports as instance attributes
    * A series of methods for socket operations:
    - send(port),
    - receive(port)
    """
    def __init__(self, ports):
        """
        Parameters: ports
        ports: a list of integers of port number
        """
        self.host = "127.0.0.1" # local host
        self.select_timeout = 0.5 # default 0.5
        self.ports = ports # input ports
        self.sending_port = ports[0] # set 1st port as the sending port
        self.ports_sockets = {} # input ports and sockets
        self.init_sockets()

    def init_sockets(self):
        """
        Parameter: ports
        ports: a list of integers of ports

        Return: port_socket
        port_socket: a list of
        """
        try:
            for port in self.ports:
                udp_socket = socket.socket(socket.AF_INET,
                                           socket.SOCK_DGRAM)
                udp_socket.bind((self.host, port))
                # udp_socket.setblocking(0)  # blocking switch

                self.ports_sockets[port] = udp_socket
        except socket.error as error:
            print("Failed to initialise sockets for ports\n", error)

    def get_ports_sockets(self):
        """
        ports_sockets getter
        """
        return self.ports_sockets

    def receive(self):
        """
        Using select() to monitor a list of ports and receive the port
        with readable data

        Parameter: sockets
        ports: a list of socket objects

        Return: (data, port)
        """
        sockets = []
        for input_socket in self.ports_sockets.values():
            sockets.append(input_socket)
        sockets_to_read = (select.select(sockets, [], [], \
                           self.select_timeout))[0]
        data_list = []
        for socket_to_read in sockets_to_read:
            # get the receiving port number which the socket binds
            # port = socket_to_read.getsockname()
            # get data from socket
            data = socket_to_read.recv(1024)
            data_list.append(data)
        return data_list

    def send(self, data_bytes, dest_port):
        """
        Parameter: data_bytes
        data_bytes: data in bytes format
        i.e. data can be the update packet from router
        """
        try:
            sending_socket = self.ports_sockets[self.sending_port]
            dest = (self.host, dest_port)
            sending_socket.sendto(data_bytes, dest)
        except KeyError:
            print("The port for sending packet does not exist")
        except socket.error as error:
            print("Can't send packet with the socket\n" + error)

    def __str__(self):
        return ("Host: {0}\n"
                "Ports: {1}\n"
                "Ports_Sockets: {2}").format(self.host,
                                             self.ports,
                                             self.ports_sockets)
```