TP5 : Programmation Réseau Année : 2022/2023

1. Compléter les deux bouts de programmes ci-dessous.
2. Tester ces deux programmes
3. Modifier le programme qui convient pour tester si les clients peuvent à leurs tours envoyer des messages multicast.
4. Tester si on peut envoyer ou recevoir des messages si on n’est pas membre du groupe multicast

import socket

import struct

import sys

message = b'very important data'

multicast\_group = ('224.3.29.71', 10000)

# Create the datagram socket.

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

# Set a timeout so the socket does not block

# indefinitely when trying to receive data.

sock.settimeout(0.2)

# Set the time-to-live for messages to 1 so they do not

# go past the local network segment.

ttl = struct.pack('b', 1)

sock.setsockopt(socket.IPPROTO\_IP, socket.IP\_MULTICAST\_TTL, ttl)

try:

 # Send data to the multicast group.

 print('sending {!r}'.format(message))

 sent = sock.sendto(message, multicast\_group)

 # Look for responses from all recipients.

 while True:

 print('waiting to receive')

 try:

 data, server = sock.recvfrom(16)

 except socket.timeout:

 print('timed out, no more responses')

 break

 else:

 print('received {!r} from {}'.format(data, server))

finally:

 print('closing socket')

 sock.close()

import socket

import struct

import sys

multicast\_group = '224.3.29.71'

server\_address = ('', 10000)

# Create the socket.

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

sock.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1)

# Bind to the server address.

sock.bind(server\_address)

# Tell the operating system to add the socket to

# the multicast group on all interfaces.

group = socket.inet\_aton(multicast\_group)

mreq = struct.pack('4sL', group, socket.INADDR\_ANY)

sock.setsockopt(socket.IPPROTO\_IP, socket.IP\_ADD\_MEMBERSHIP, mreq)

# Receive/respond loop

while True:

 print('\nwaiting to receive message')

 data, address = sock.recvfrom(1024)

 print('received {} bytes from {}'.format(len(data), address))

 print(data)

 print('sending acknowledgement to', address)

 sock.sendto(b'ack', address)