Lucie MICHELET
ELSS – 09/02/23

PW1

2. UDP

Sender :

```python
import socket

UDP_IP = "127.0.0.1"
UDP_PORT = 43210
MESSAGE = "Hello, World!"   # using utf8 char

print("UDP target IP:", UDP_IP)
print("UDP target port:", UDP_PORT)

sock = socket.socket(socket.AF_INET, # Internet
                     socket.SOCK_DGRAM) # UDP
sock.sendto(MESSAGE.encode('utf8'), (UDP_IP, UDP_PORT))
```

Receiver :

```python
import socket

IP   = "127.0.0.1"
PORT = 43210

sock = socket.socket(socket.AF_INET, # Internet
                     socket.SOCK_DGRAM) # UDP

print("trying to bind to ", IP,PORT)
sock.bind((IP,PORT))# association de la socket avec IP/PORT
print("waiting for data....")

while True:
    data,addr=sock.recvfrom(1024) # atte
    print("received message:", data.decode('utf8'))
    print("from:",addr)
```

Result in receiver console :

```
trying to bind to  127.0.0.1 43210
waiting for data....
received message: Héllo, World!
from: ('127.0.0.1', 50245)
```

Lucie MICHELET
ELSS – 09/02/23

Changing network and *IP* :

```
Carte réseau sans fil Wi-Fi :

   Suffixe DNS propre à la connexion. . . :
   Adresse IPv6 de liaison locale. . . . .: fe80::e71e:a32d:b987:c528%9
   Adresse IPv4. . . . . . . . . . . . . .: 192.168.1.26
```

```
IP   = "192.168.1.26"
PORT = 52001
```

What we receive from port 52001 then :

```
from: ('192.168.1.87', 62908)
received message:
48.813728671885634,2.3934930846866003,105,5,1415263338000

from: ('192.168.1.87', 62908)
received message:
"latitude","longitude","altitude","accuracy","time"

from: ('192.168.1.87', 62908)
received message:
48.81121270075116,2.3841758263391872,58,35,1415262788000
```

What we receive from port 52002 :

```
UnicodeDecodeError: 'utf-8' codec can't decode byte 0xe0 in
position 1: invalid continuation byte
```

Decoding data from bytes :

```
#%%
while True:
    data,addr=sock.recvfrom(1024) # attente d'un message de lecture
    recieved_bytes = data
    format = 'diiiiffff' # byte array format : d for double , i for int, f for float

    decoded = struct.unpack(format,recieved_bytes[0:40])
    print("from:",addr)
    print(decoded)
```

```
trying to bind to  192.168.1.26 52002
waiting for data....
from: ('192.168.1.87', 51820)
(1629293623.7980149, 62, 83, 60, 44, 103.6500015258789,
-28.0, -30.0, -987.0)
from: ('192.168.1.87', 51820)
(1629293623.900185, 62, 84, 60, 44, 103.69999694824219,
-40.0, -30.0, -994.0)
from: ('192.168.1.87', 51820)
(1629293624.003519, 62, 83, 60, 44, 103.68000030517578,
-37.0, -27.0, -983.0)
```

Lucie MICHELET
ELSS – 09/02/23

Display date :

```
import datetime
print(datetime.datetime.fromtimestamp(decoded[0]))
```

```
2021-08-18 15:33:44.513046
```

The result when we had that line in the while :

```
from: ('192.168.1.87', 51820)
date : 2021-08-18 15:33:47.929971
(1629293627.929971, 62, 79, 60, 44, 103.62999725341797, -44.0, -23.0, -1001.0)

from: ('192.168.1.87', 51820)
date : 2021-08-18 15:33:48.031889
(1629293628.031889, 62, 78, 60, 44, 103.58999633789062, -31.0, -15.0, -1004.0)
```

Select :

```
import select

#%%

IP   = "192.168.1.26"
PORT1 = 52001
PORT2 = 52002

sock = socket.socket(socket.AF_INET, # Internet
                     socket.SOCK_DGRAM) # UDP
sock1 = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock2 = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

print("trying to bind socket 1 to ", IP,PORT1)
sock1.bind((IP,PORT1))
print("trying to bind socket 2 to ", IP,PORT2)
sock2.bind((IP,PORT2))
print("sockets binded ! ")

#%%
socketList = [sock1,sock2]
while True:
    ls = select.select(socketList,[],[],1)
    for s in ls[0]:
        data, addr = s.recvfrom(1024) # buffer size is 1024 bytes
        print("#message from:",addr)
        print("#message to :",s.getsockname())
        print("#message :", data) # depending on getsockname you should adapt # your decoding method
```

Lucie MICHELET
ELSS – 09/02/23

Result :

```
In [13]: runcell(1, 'C:/Users/Lucie/Desktop/COURS/AERO4/S2/SYSTEME/
RESEAUX/PW01_etu/UDP_RECV_1.py')
trying to bind socket 1 to  192.168.1.26 52001
trying to bind socket 2 to  192.168.1.26 52002
sockets binded !

In [14]: runcell(3, 'C:/Users/Lucie/Desktop/COURS/AERO4/S2/SYSTEME/
RESEAUX/PW01_etu/UDP_RECV_1.py')
#message from: ('192.168.1.87', 62908)
#message to : ('192.168.1.26', 52001)
#message : b'48.81460554347133,2.3930909128510702,90,10,1415263260000\n'
#message from: ('192.168.1.87', 51820)
#message to : ('192.168.1.26', 52002)
#message : b'\xce;\xd9\x16CG\xd8AC\x00\x00\x00W\x00\x00\x00F\x00\x00\x00%
\x00\x00\x00\x00\x00\xcfB\x00\x00\xf8\xc1\x00\x00\xf0A\x00\x80{\xc4'
#message from: ('192.168.1.87', 62908)
#message to : ('192.168.1.26', 52001)
#message : b'48.81449964257766,2.3934848755146576,84,10,1415263275000\n'
```

```
#message from: ('192.168.1.25', 49993)
#message to : ('192.168.1.26', 52002)
#message : b'B\xe8\x83\x19CG\xd8AD\x00\x00\x00_\x00\x00\x00F\x00\x00\x00 \x00\x00\x00q
\xbd\xceB\x00\x00\x92\xc2\x00\x00\xa0\xc0\x00@t\xc4'

#message from: ('192.168.1.87', 51820)
#message to : ('192.168.1.26', 52002)
#message : b'\xb9\xbe\x92\x11CG\xd8A@\x00\x00\x00\n\x00\x00\x00\x00\x00\x00\x00*
\x00\x00\x00R\xb8\xcdB\x00\x00\x94B\x00\x00\xb6B\x00\xc0^\xc4'

#message from: ('192.168.1.87', 62908)
#message to : ('192.168.1.26', 52001)
#message : b'48.81292958291009,2.387834826761983,86,15,1415262983000\n'

#message from: ('192.168.1.25', 49993)
#message to : ('192.168.1.26', 52002)
#message : b'\ri\x8a\x19CG\xd8AD\x00\x00\x00^\x00\x00\x00F\x00\x00\x00
\x00\x00\x00\xf6\xa8\xceB\x00\x00\xb6\xc2\x00\x00\xa0@\x00\x00u\xc4'
```

We can see that the highest speed rate is port **52002**.

Lucie MICHELET
ELSS – 09/02/23

3. client server TCP / IP using python

What is the 127.0.0.1 IP address? What is-it used for ?

The 127.0.0.1 address is the local host. Also called loopback, it's used to work on the LAN.

The source code of TCP client and server are available on the annex. According to your knowledge, which code corresponds to the client and which one to the server? Why ?

Server/Reciever :

```python
import socket

TCP_IP = '127.123.234.1'
TCP_PORT = 50005
BUFFER_SIZE = 1024

sconn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

sconn.bind((TCP_IP, TCP_PORT))
sconn.listen(1)

s, addr = sconn.accept()
rawdata = s.recv(BUFFER_SIZE)
print("received data:", rawdata.decode('ascii'))
s.send(rawdata)  # echo

s.close()
sconn.close()
```

Client/Sender :

```python
import socket

TCP_IP = '127.123.234.1'
TCP_PORT = 50005
BUFFER_SIZE = 1024                    # read size
msg = "Hello, Everyone!"

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((TCP_IP, TCP_PORT))

print('sending data...')
s.send(msg.encode('ascii'))

rawdata = s.recv(BUFFER_SIZE)
print("received data:\n", rawdata.decode('ascii'))

s.close()
```

On server side two sockets are used. Explain why :

On the server side, two sockets are used because one assure the connection between the two machine and accept the communication. The second socket receives the data and allows communication.

Lucie MICHELET
ELSS – 09/02/23

Code

Server :

```python
import socket
TCP_IP = '127.0.0.1'
TCP_PORT = 55000
BUFFER_SIZE = 1024

sconn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

print("binding to "+ TCP_IP +":"+str(TCP_PORT)," ...")
sconn.bind((TCP_IP, TCP_PORT))

print("Waiting client ...")
sconn.listen(1)
s, addr = sconn.accept()
print('Client connected with address:', addr)

rawdata = s.recv(BUFFER_SIZE)

print("received data:", rawdata.decode('ascii'))

s.send(rawdata) # echo

s.close()
sconn.close()
```

Client :

```python
import socket

TCP_IP = '127.0.0.1'
TCP_PORT = 55000
BUFFER_SIZE = 1024 # read size

msg = "Hello, Everyone!"
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

print('connecting to '+ TCP_IP + ':' + str(TCP_PORT) +'...')
s.connect((TCP_IP, TCP_PORT))
print('sending data...')

s.send(msg.encode('ascii'))
rawdata = s.recv(BUFFER_SIZE)
print("received data:\n", rawdata.decode('ascii'))

s.close()
```

Lucie MICHELET
ELSS – 09/02/23

Result

Server :

```
Desktop/COURS/AERO4/S2/SYSTEME/RESEAUX/TP1 )
binding to 127.0.0.1:55000  ...
Waiting client ...
Client connected with address: ('127.0.0.1', 54014)
received data: Hello, Everyone!
```

Client :

```
connecting to 127.0.0.1:55000...
sending data...
received data:
 Hello, Everyone!
```

Mini Chat

Server :

```python
import socket
TCP_IP = '10.9.127.207'
#TCP_IP = '127.0.0.1'
TCP_PORT = 55000
BUFFER_SIZE = 1024

sconn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

print("binding to "+ TCP_IP +":"+str(TCP_PORT)," ...")
sconn.bind((TCP_IP, TCP_PORT))

print("Waiting client ...")
sconn.listen(1)
s, addr = sconn.accept()
print('Client connected with address:', addr)

D = '';
while D != 'fin':
    data = s.recv(BUFFER_SIZE)
    name = socket.gethostname()
    print("\nClient ",name,":\n", data.decode('ascii'))
    if data.decode('ascii') == 'fin':
        break
    data = input("Serveur :\n").encode('ascii')
    s.send(data)
    print('sending data...\n')
    D = data.decode('ascii')
```

Lucie MICHELET
ELSS – 09/02/23

Client :

```python
import socket

TCP_IP = '127.0.0.1'
TCP_PORT = 55000
BUFFER_SIZE = 1024 # read size

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

print('connecting to '+ TCP_IP + ':' + str(TCP_PORT) +'...')
s.connect((TCP_IP, TCP_PORT))


D = '';

while D != 'fin':
    data = input("Client : \n") .encode('ascii')
    print('sending data...\n')
    s.send(data)
    if data.decode('ascii') == 'fin':
        break
    data = s.recv(BUFFER_SIZE)
    print("Serveur :\n", data.decode('ascii'))
    D = data.decode('ascii')


s.close()
```

Result :

```
binding to 10.9.127.207:55000  ...
Waiting client ...
Client connected with address: ('10.9.127.246', 61640)

Client  LAPTOP-HF4VBLD3 :
 slt

Serveur :
oui
sending data...


Client  LAPTOP-HF4VBLD3 :
 fin
```

Lucie MICHELET
ELSS – 09/02/23

4 Micro HTTP server

```
import time
import socket

#TCP_IP = '10.9.127.207'
TCP_IP = '127.0.0.1'
TCP_PORT = 55000
BUFFER_SIZE = 1024
```

```
#%% Fabien est le plus fort !

http = b"HTTP/1.1 200 OK\n\
Date: Sun, 29 Mar 2015 10:48:13 GMT\n\
Expires: -1\n\
Cache-Control: private, max-age=0\n\
Content-Type: text/html;\n\
charset=ISO-8859-1\n\n"
html= b"<html><body><h1>Fabien c'est le  plus fort</h1>\
</body></html>\n\n"

http_response = http+html
```

```
sconn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

print("binding to "+ TCP_IP +":"+str(TCP_PORT)," ...")
sconn.bind((TCP_IP, TCP_PORT))

print("Waiting client ...")
sconn.listen(1)

while True:
    s, addr = sconn.accept()
    print('Client connected with address:', addr)

    data = s.recv(BUFFER_SIZE)
    s.send(http_response)
    print('sending data...\n')
    s.close()

print('Socket closed.')
sconn.close()
```