

CS-GY 6843

Wireshark Socket Programming 2

03/22/2022

Yibei Huang

```
Command Prompt
(c) Microsoft Corporation. All rights reserved.
C:\Users\huang>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 3:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::f4c6:a3fc:2281:4c2f%18
    IPv4 Address. . . . . : 192.168.31.230
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.31.1
```

```

UDPClientSocket = socket(family=AF_INET, type=SOCK_DGRAM)
UDPClientSocket.settimeout(1)
server_addr = (host, port)
resps = []

```

1. Create the client socket and set the type as UDP.
2. Set the time out to 1 second.
3. Create server address and port tuple.
4. Create an empty list to store future responds from server.

```

try:
    start_time = time.time()
    message = "Ping " + str(seq) + " " + str(start_time)
    UDPClientSocket.sendto(message.encode(), server_addr)
    data, server = UDPClientSocket.recvfrom(4096)
    resps.append((seq, data.decode(), float(data.decode().split()[3]) - float(data.decode().split()[2])))
    # Fill in end
except timeout:
    resps.append((seq, "Request timed out", 0))
    print("Ping " + str(seq) + " request timed out")

```

1. For each iteration, record client's send time
2. Create message to send to the server
3. Send the message through the socket
4. Receive responds from the server
5. Store sequence number, server respond and calculated rtt to the list if got a respond
6. If timed out, store the timed out information in the list.

The extra credit Ping Statistics was completed in file <client_exercise1.py>.