

חלק ג - Part c

What is CRC?

CRC stands for "Cyclic Redundancy Check". It's a way to verify if the data has been corrupted in transit. The bits of the data are interpreted as a long polynomial, and the remainder of this long polynomial divided by a prime polynomial is the magic number that can tell whether or not the data has been corrupted or not. If the calculated number matches the number in the packet, it is assumed that the packet was not corrupted.

What is the difference between QUIC, HTTP 1.0, HTTP 1.1 and HTTP 2.0?

HTTP 1.0 is the first protocol of HTTP. It is sequential and one connection leads to one transfer. It is TCP based, text-based.

HTTP 1.1 is almost the same as HTTP 1.0, but it has the required **Host** header.

HTTP 2.0 has support for multiple transfers triggered by one connection. This is useful in a web environment, where websites are made of a significant amount of files.

QUIC is a UDP-based, binary-based protocol that has less overhead than HTTP. It is experimental as of 2021.

Why we need port numbers?

A computer has only one address in the network, but multiple applications on it might need connectivity. Port numbers allow for multiple applications in a computer to have connectivity while sharing the same address.

What is subnet and why is it necessary?

A subnet is a network inside a network. A home has only one router, which has only one address to the Internet, but the home might have multiple devices (smartphones, computers) that need to connect to the Internet. Devices in the subnet communicate with the gateway and, in doing so, are able to communicate with the Internet.

A subnet can also be useful as a security measure, where the gateway (the node that connects the subnet to the Internet) acts as a firewall.

Why we need MAC addresses, why is it not enough to work just with the IP address?

IP addresses can be spoofed, enabling one computer to impersonate another computer. However, the hardware address (MAC) prevents that from happening to some extent.

What is the difference between Router, Switch and NAT?

Router is a device that creates a local area network (subnet) and manages the traffic that goes through it.

Switch is a device that connects various devices into a single computer network.

NAT (Network Address Traversal) is a protocol that allows for connections from the Internet to a computer inside a subnet to be established. A table that keeps tracks of the live connections is stored in the router and that is used to decide where to forward which packets.

What are the methods to overcome the IPv4 shortage?

One of them is to simply use the IPv6 protocol. Other methods are using subnets with NAT and port forwarding.

Given the following network:

AS2, AS3 runs OSPF

AS1, AS4 runs RIP

Between the ASx a BGP is running.

With which protocol the router X learns about subnet Y?

AS1 TO AS4, AS4 TO AS1: Distance vector

AS1, AS4 TO AS2, AS3: Distance vector

AS2 TO AS3, AS3 TO AS2: Shortest path first

PYTHON CHAT APP

GETTING STARTED

In order to use this chat application, the server that handles the communication must be set up first. This can be done with the following command at the root folder of the project:

```
python3 main-server.py
```

```
sh-5.0$ ls
downloads  main-client.py  network    tests.py
links.txt  main-server.py  README.md  uploads
sh-5.0$ python main-server.py
```

The server will silently run in the background. After the server is running, users start the chat application with the following command:

```
python3 main-client.py
```

```
sh-5.0$ ls
downloads  main-client.py  network  tests.py
links.txt  main-server.py  README.md uploads
sh-5.0$ python3 main-client.py
```

Python Chat App

USER NAME:

NOT LOGGED IN!

NEW MESSAGE:

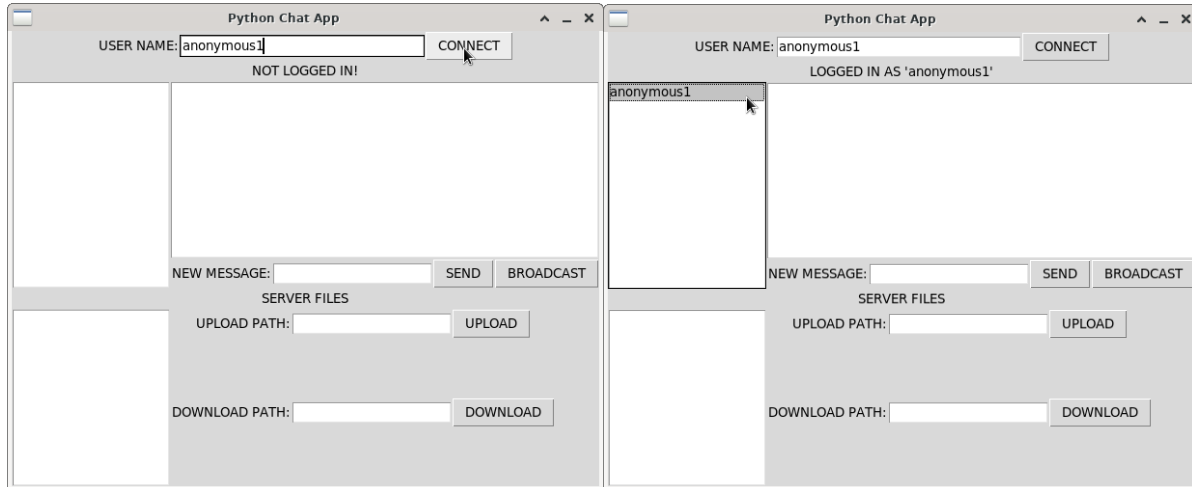
SERVER FILES

UPLOAD PATH:

DOWNLOAD PATH:

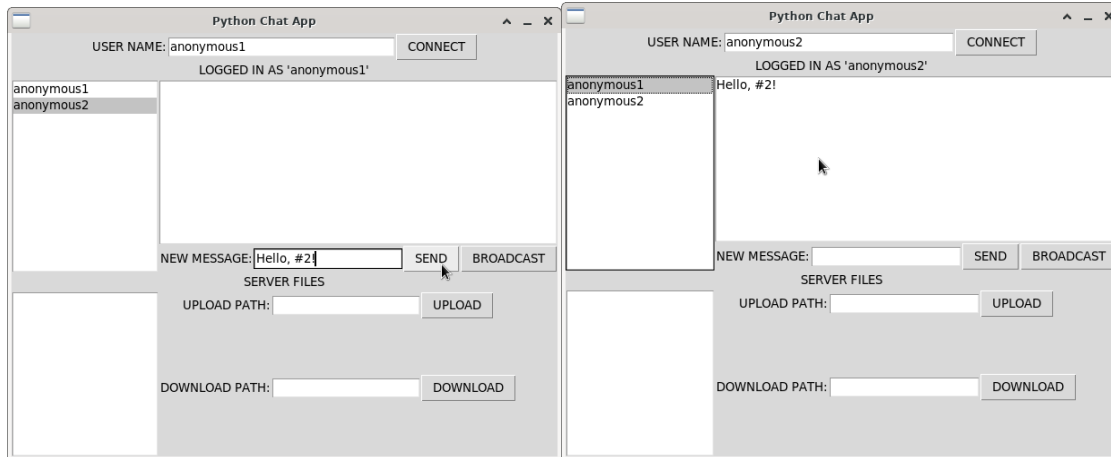
CONNECTING TO SERVER

After the server application has started and the client application has successfully launched, users must pick an username and click the 'CONNECT' button in the client app. After a successful log-in, a message showing that the user has indeed logged-in will be presented and the user name will be added to the list of online users (listbox at the upper left).



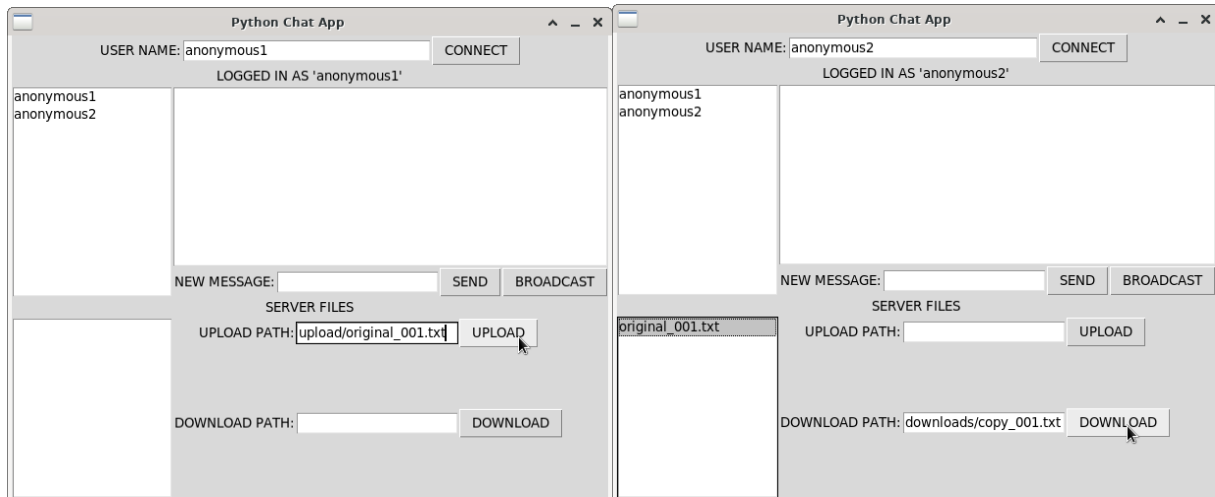
שליחת הודעות – MESSAGE COMMUNICATION

In order to send a message to another user, click the target user name in the online users listbox, write the message and click 'SEND'. The target user can check received messages by clicking the user name of the source user.



FILE UPLOAD AND DOWNLOAD

In order to upload a file, a user must write in the 'UPLOAD PATH' entry box the path for the file to be uploaded and click the 'UPLOAD' button. After a successful upload, the name of the file will be displayed in the files list provided by the server (listbox at the lower left corner). A user wishing to download one of these files must click the target file, write the path for the downloaded file and click the 'DOWNLOAD' button.



חלק ב - Part B

HOW THE SYSTEM MANAGES LOST PACKETS:

All packets are numbered, and every request from the client must be followed by a reply from the server. If the server fails to reply, the client sends the packet again and again, until it either receives a reply from the server acknowledging the packet sent or gives up because too much time has elapsed.

HOW THE SYSTEM OVERCOMES LATENCY:

The size of the packets are minimal, below 508 bytes, in order to minimize the possibility of a packet being dropped. Less dropped packets leads to a lower latency.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	UDP	46	56417 → 55000 Len=14
2	0.000526	127.0.0.1	127.0.0.1	UDP	50	55000 → 56417 Len=18
3	3.465625	127.0.0.1	127.0.0.1	UDP	42	56417 → 55000 Len=10
4	3.466111	127.0.0.1	127.0.0.1	UDP	44	55000 → 56417 Len=12
5	3.605543	127.0.0.1	127.0.0.1	UDP	50	56417 → 55000 Len=18
6	3.605941	127.0.0.1	127.0.0.1	UDP	54	55000 → 56417 Len=22
7	3.730821	127.0.0.1	127.0.0.1	UDP	42	56417 → 55000 Len=10
8	3.731467	127.0.0.1	127.0.0.1	UDP	36	55000 → 56417 Len=4
9	3.855487	127.0.0.1	127.0.0.1	UDP	42	56417 → 55000 Len=10
10	3.855851	127.0.0.1	127.0.0.1	UDP	36	55000 → 56417 Len=4
11	4.121224	127.0.0.1	127.0.0.1	UDP	42	56417 → 55000 Len=10

> Frame 1: 46 bytes on wire (368 bits), 46 bytes captured (368 bits)

> Null/Loopback

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

> User Datagram Protocol, Src Port: 56417, Dst Port: 55000

> Data (14 bytes)

PACKET TYPE #1

USERNAME

USERNAME LENGTH

0000	02 00 00 00 45 00 00 2a 54 68 00 00 80 11 00 00E...* Th.....
0010	7f 00 00 01 7f 00 00 01 dc 61 d6 d8 00 16 21 89a....!..
0020	00 01 00 0a 51 6e 6f 6e 79 6d 6f 75 73 31anon ymous1

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	UDP	46	56417 → 55000 Len=14
2	0.000526	127.0.0.1	127.0.0.1	UDP	50	55000 → 56417 Len=18
3	3.465625	127.0.0.1	127.0.0.1	UDP	42	56417 → 55000 Len=10
4	3.466111	127.0.0.1	127.0.0.1	UDP	44	55000 → 56417 Len=12
5	3.605543	127.0.0.1	127.0.0.1	UDP	50	56417 → 55000 Len=18
6	3.605941	127.0.0.1	127.0.0.1	UDP	54	55000 → 56417 Len=22
7	3.730821	127.0.0.1	127.0.0.1	UDP	42	56417 → 55000 Len=10
8	3.731467	127.0.0.1	127.0.0.1	UDP	36	55000 → 56417 Len=4
9	3.855487	127.0.0.1	127.0.0.1	UDP	42	56417 → 55000 Len=10
10	3.855851	127.0.0.1	127.0.0.1	UDP	36	55000 → 56417 Len=4

> Frame 2: 50 bytes on wire (400 bits), 50 bytes captured (400 bits)
 > Null/Loopback
 > Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
 > User Datagram Protocol, Src Port: 55000, Dst Port: 56417
 > Data (18 bytes)

PACKET #2 SESSION COOKIE

USER NUMBER
SESSION NUMBER

0000	02 00 00 00 45 00 00 2e	54 69 00 00 80 11 00 00E...Ti....
0010	7f 00 00 01 7f 00 00 01	d6 d8 dc 61 00 1a a2 21a...!
0020	00 02 20 4a 1a fa 08 f1	2e a8 c5 63 36 e9 99 ad	..J....c6...
0030	a3 81		..

EXPLANATION FOR CLASSES

ServerProtocol:

class that will be used to create a datagram socket for the server using
loop.create_datagram_endpoint

Server:

class that handles the server state (files, sessions)

ClientProtocol:

class that will be used to create a datagram socket for the client using
loop.create_datagram_endpoint

Client:

client that manages the state of the client (session, online users)