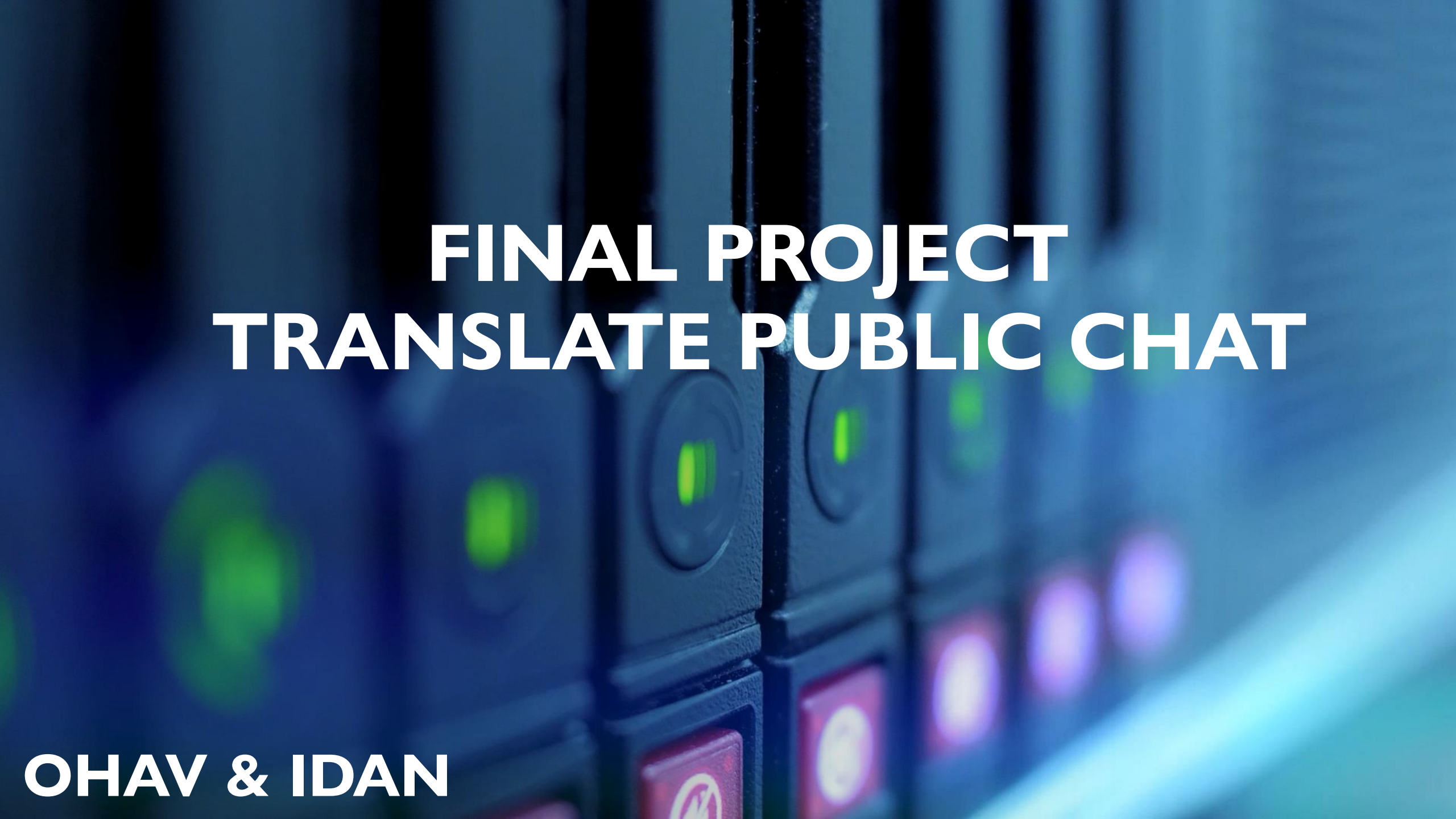


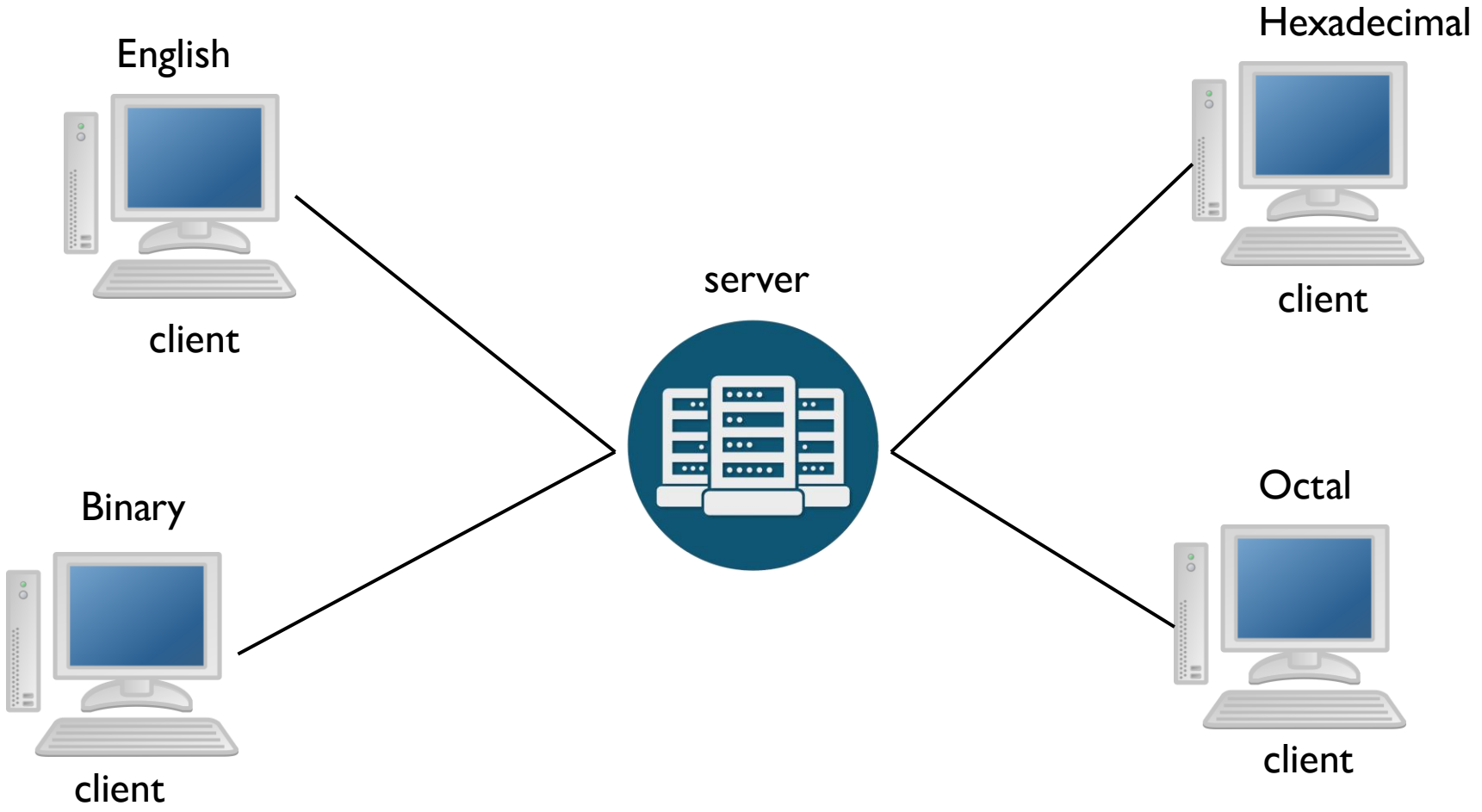
FINAL PROJECT
TRANSLATE PUBLIC CHAT

OHAV & IDAN

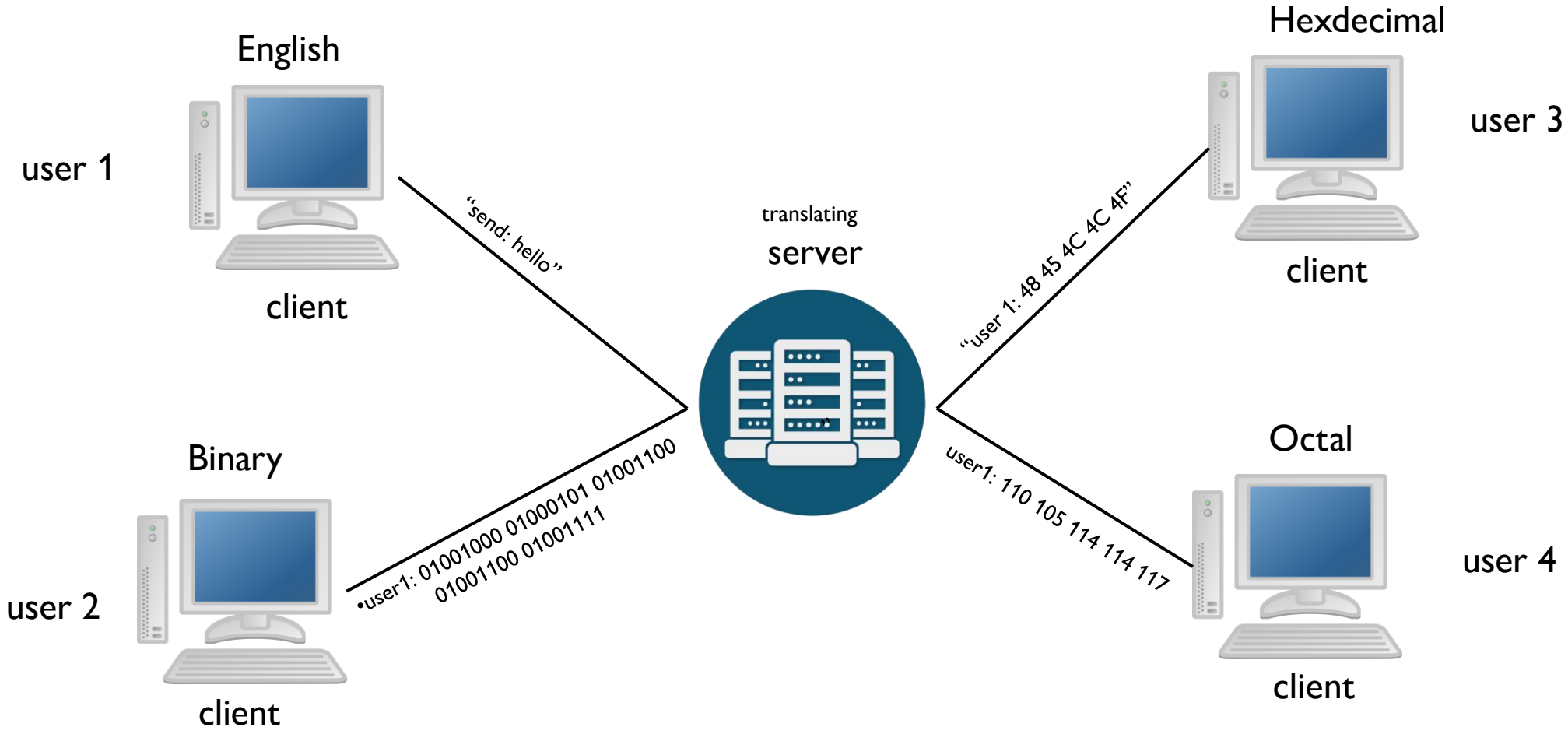
Main Idea

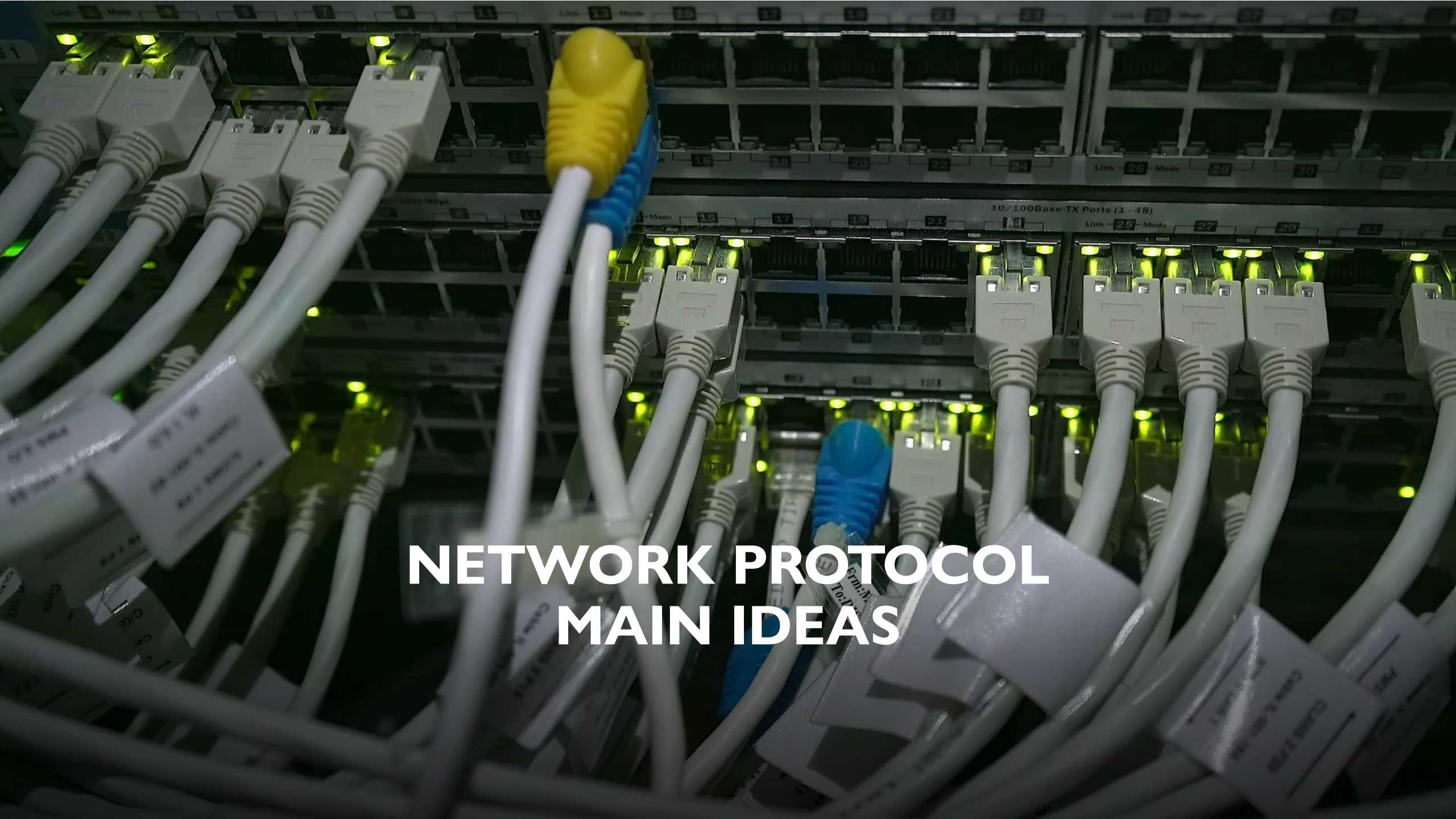
- Public chat system with multiple clients
- Each client uses a unique language
- Messages are automatically translated
- Each client reads messages in their own language

Client-Server Model



Demonstration Of Application

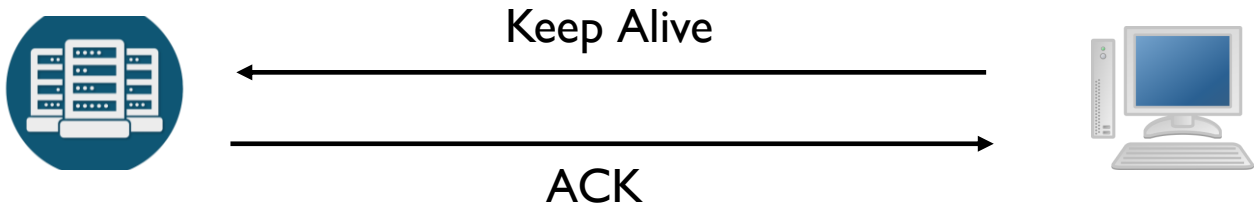




NETWORK PROTOCOL MAIN IDEAS

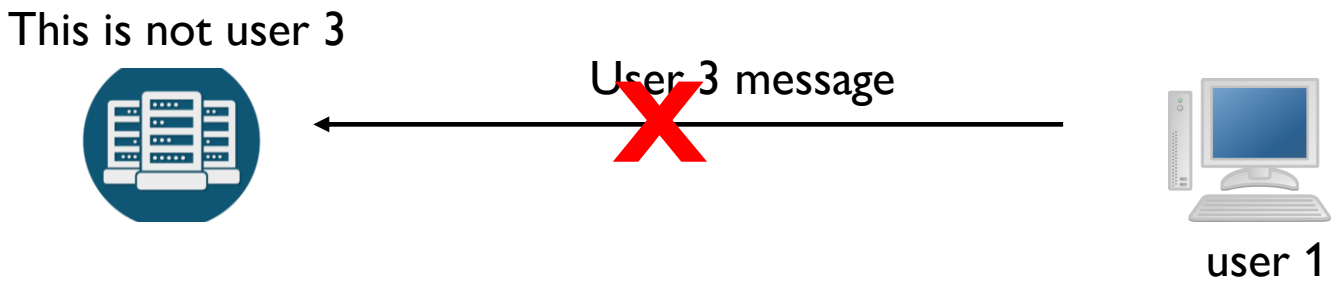
Keep Alive And ACK

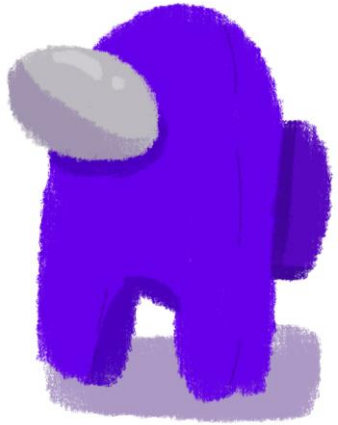
- Keep-alive with the server, each 30 seconds.
- Valid for 60 seconds – half of time
- check if there is connection and the connection is stable
- ACK response from server



Authentication

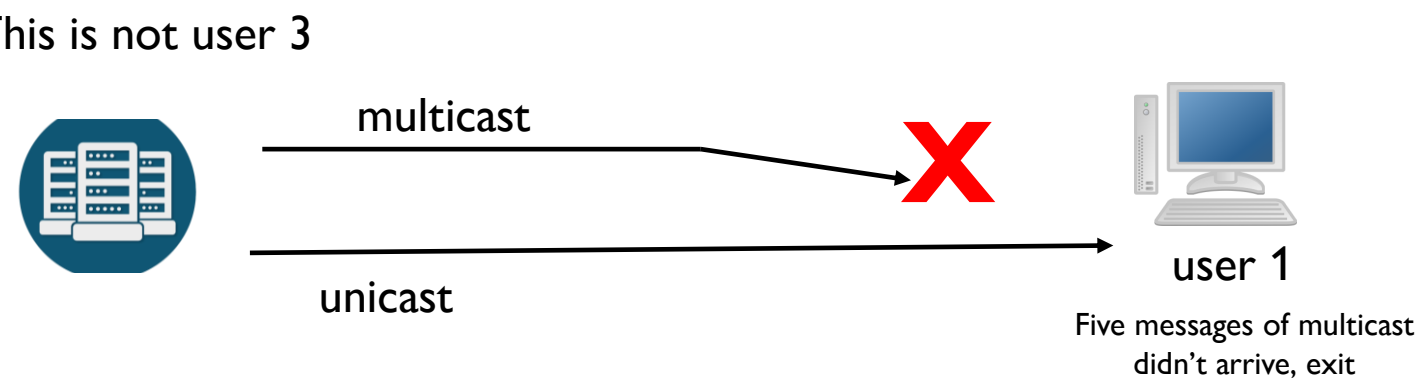
- Check if it's the real client using authentication.
- Comparing by ID
- If not, disconnect the client from chat.





Check if Multicast Message Arrived

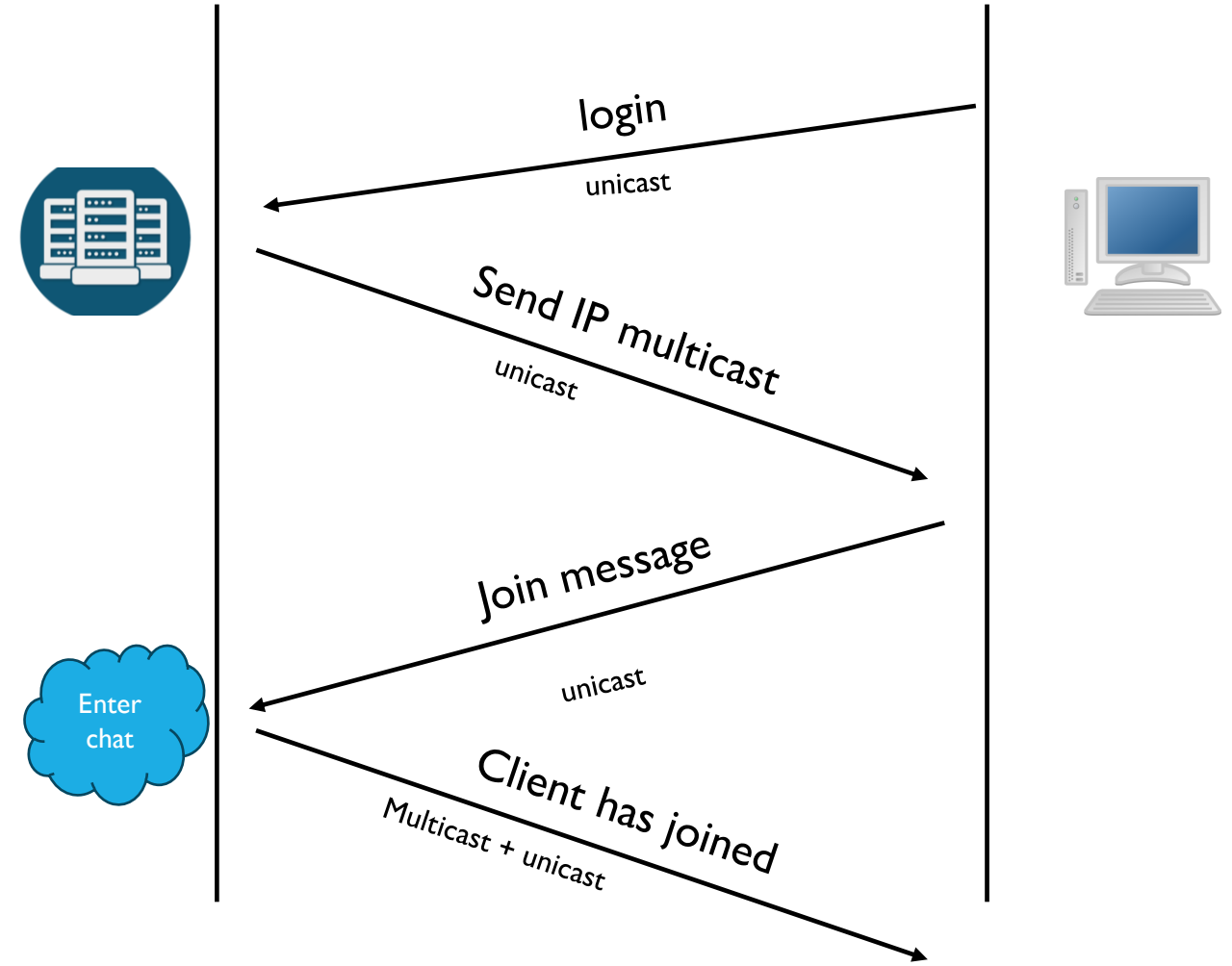
- Unicast send alongside multicast
- Maintain multicast message reliability
- If more than 5 multicast massages fall in a row, client close socket.
- Note:
  multicast on UDP and unicast is on TCP,



Establish

When client connect to server the protocol is in establish state .

- The client sent login message
- verify, send the multicast ip (chat entrance)
- The client send Loggin confirm message (Join)
- The server add the client to users mangment
- The server send new join multicast



Protocol Messages

- Struct message : unicast message

Type	ID	Language	Size-Payload	Payload
Type of command	ID number	Language number	Size of payload	Message data

- Struct multicast message :

Binary payload	Hexadecimal payload	English payload	Octal payload
Message in binary translated	Message in hexadecimal translated	Message in English translated	Message in octal translated

* If ther is no client in the room that speech specific language, the unused language field is empty

Server commands

Command	No.	Porpus
Login	1	accept registeration data of client
JOIN_Message	3	accept join confirm of the client
Send_message	4	send meesage to chat group
Exit	6	accept Exit and send ack_exit message
Keep_alive	7	accept keep alive message
No_LOG	9	accept message indicate trouble in the establishment

Client commands

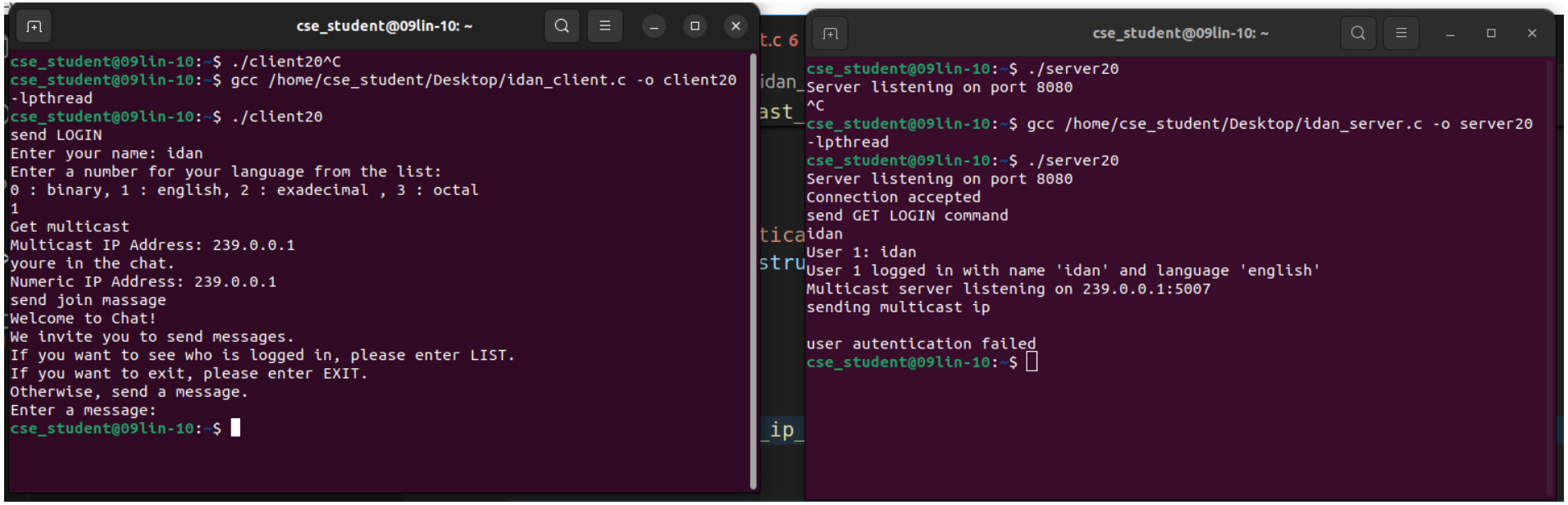
Command	No.	Porpus
multicast_indicator	10	multicast message didn't arrive
Authentication	9	failed authenticate user
Get_List	5	get list of online
Get_Multicast	2	get multicast message
Ack_exit	8	disconnect
LOGIN	1	login

QA

```
case Get_Multicast:
    printf("Get multicast\n");

    parse_multicast_ip_and_join(recv.payload);
    struct_message join;
    //join.id = client_struct.id;
    join.id  = 3;
    join.laguage = language;
```

Authentication

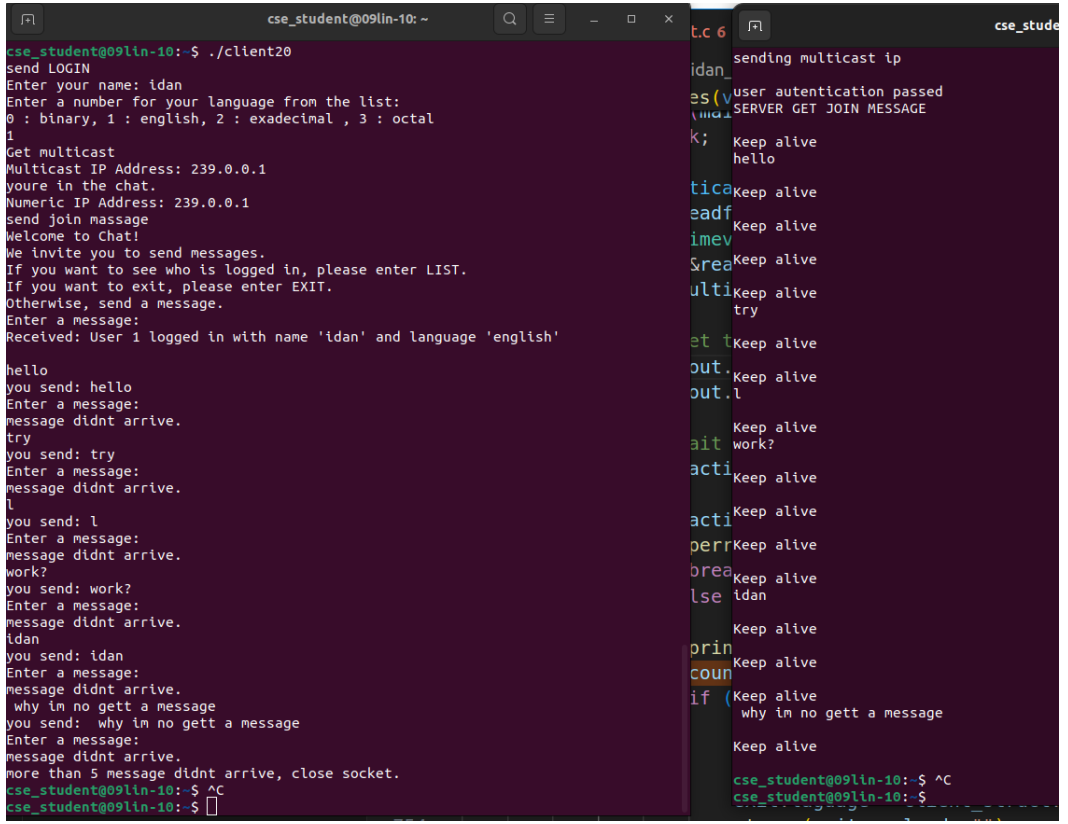
- An imposter client was manually set to be client ID 3.
- The server detected the impersonation attempt.
- An authentication issue was raised by the server.



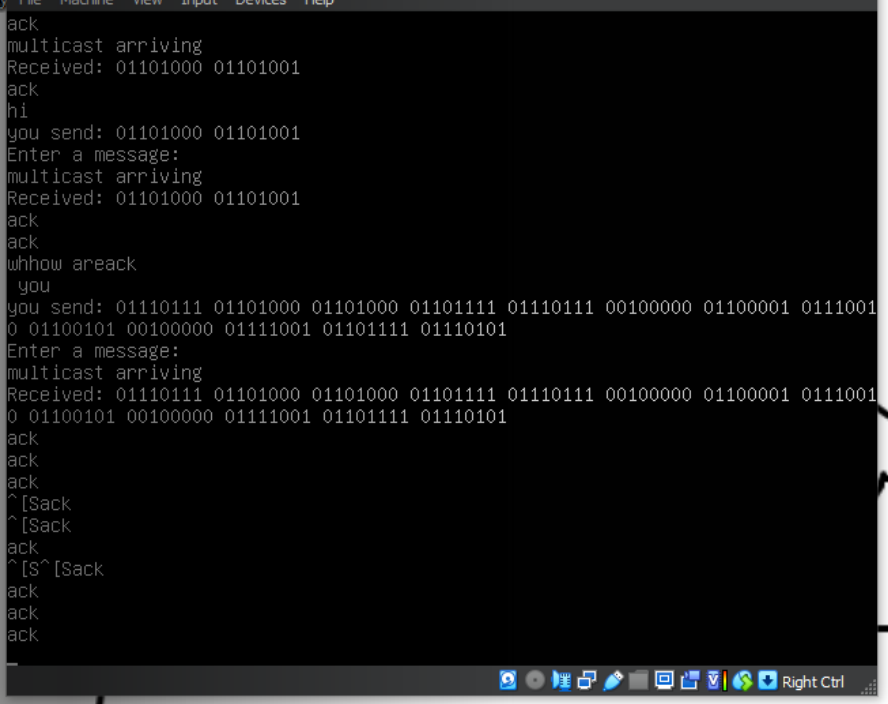
Multicast Message Loss

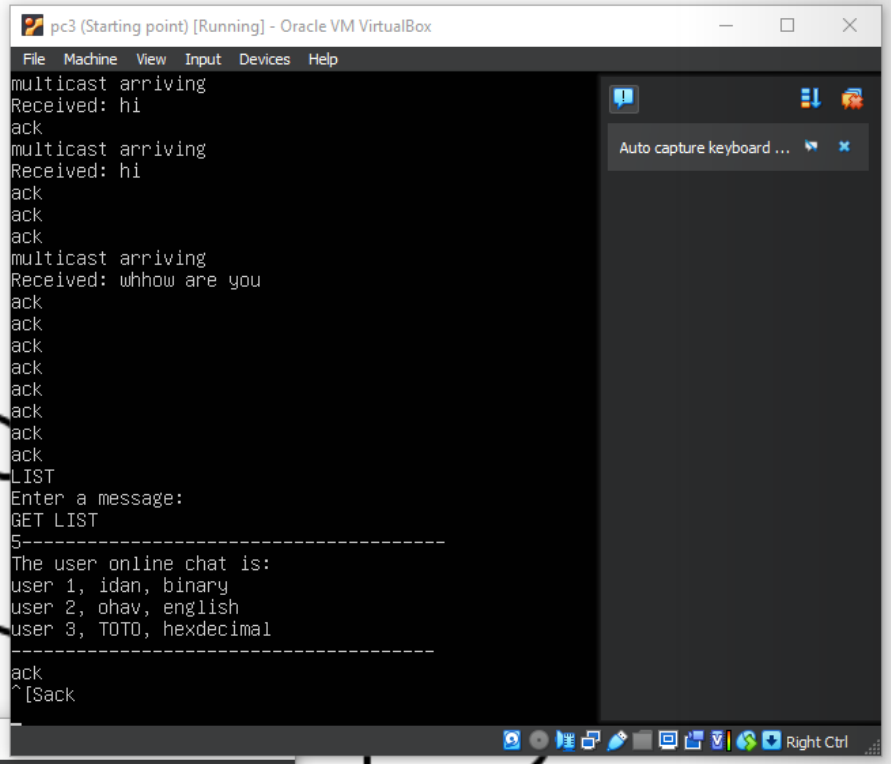
```
    }

    }
    sleep(0.5);
    //disable multicast send message
    // if (sendto(mult.sockfd, &send, sizeof(send), 0, (struct sockaddr *)&mult.multicast_addr, sizeof(mult.multicast_addr)) < 0) {
    //     perror("sendto");
    // }
```

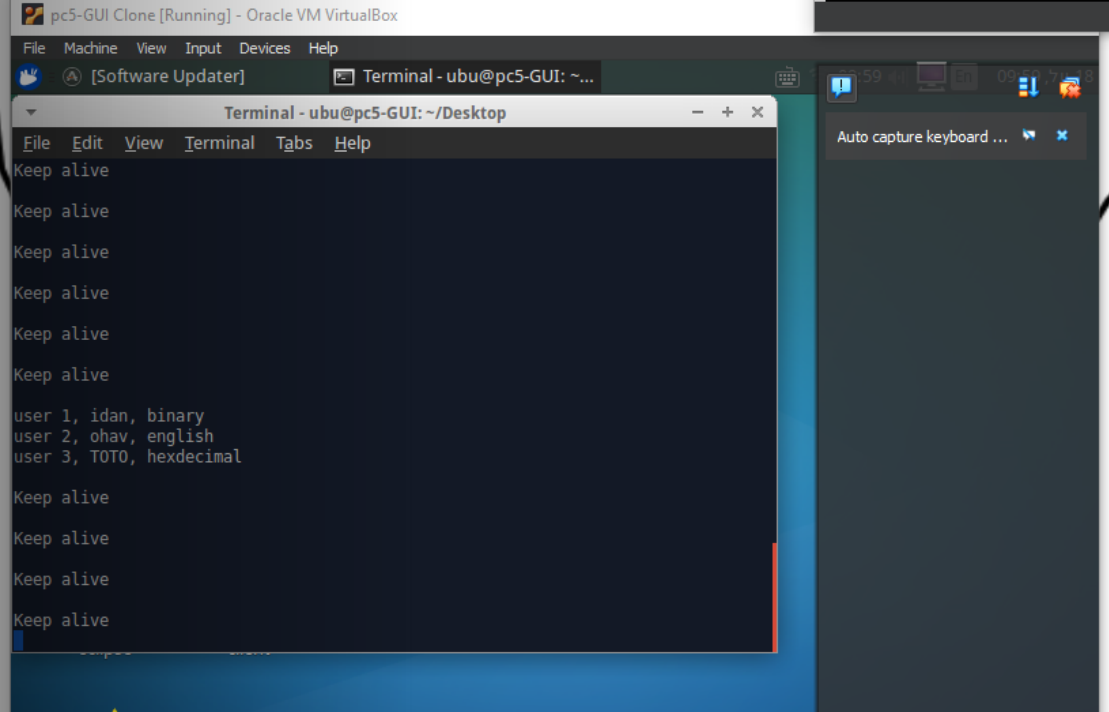
- Multicast was disabled to observe protocol handling
- After sending 5 multicast messages, the socket was closed

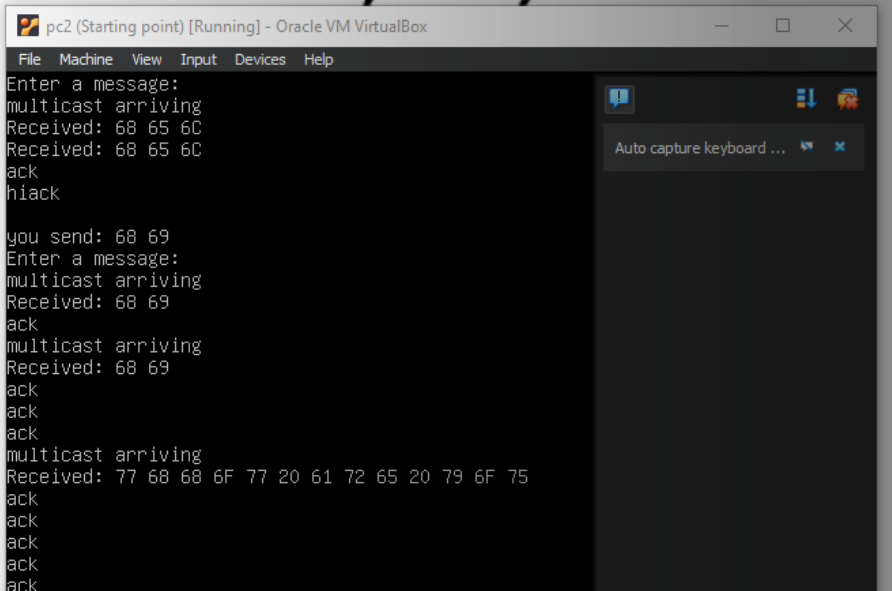


DEMONSTRATION

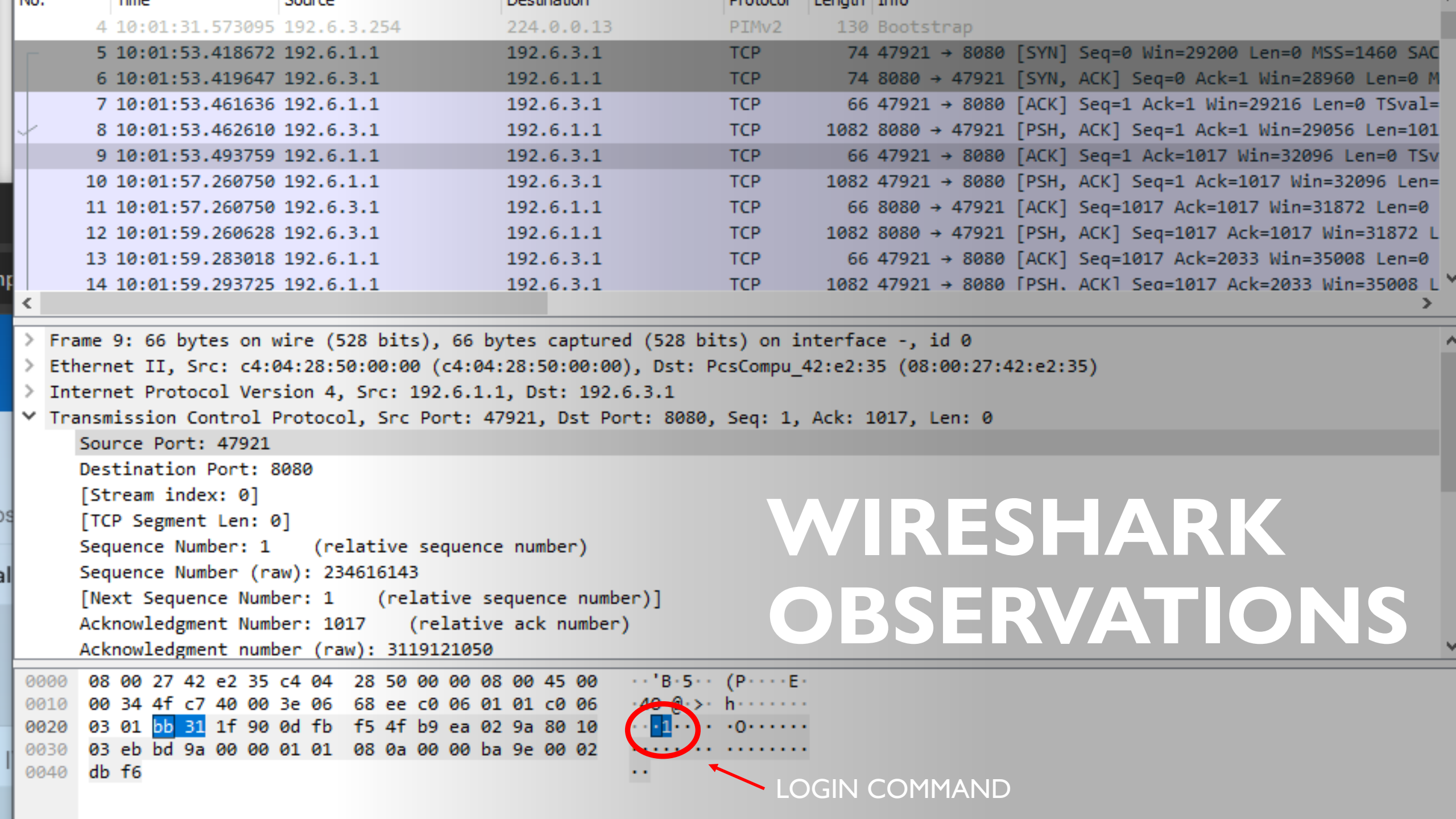




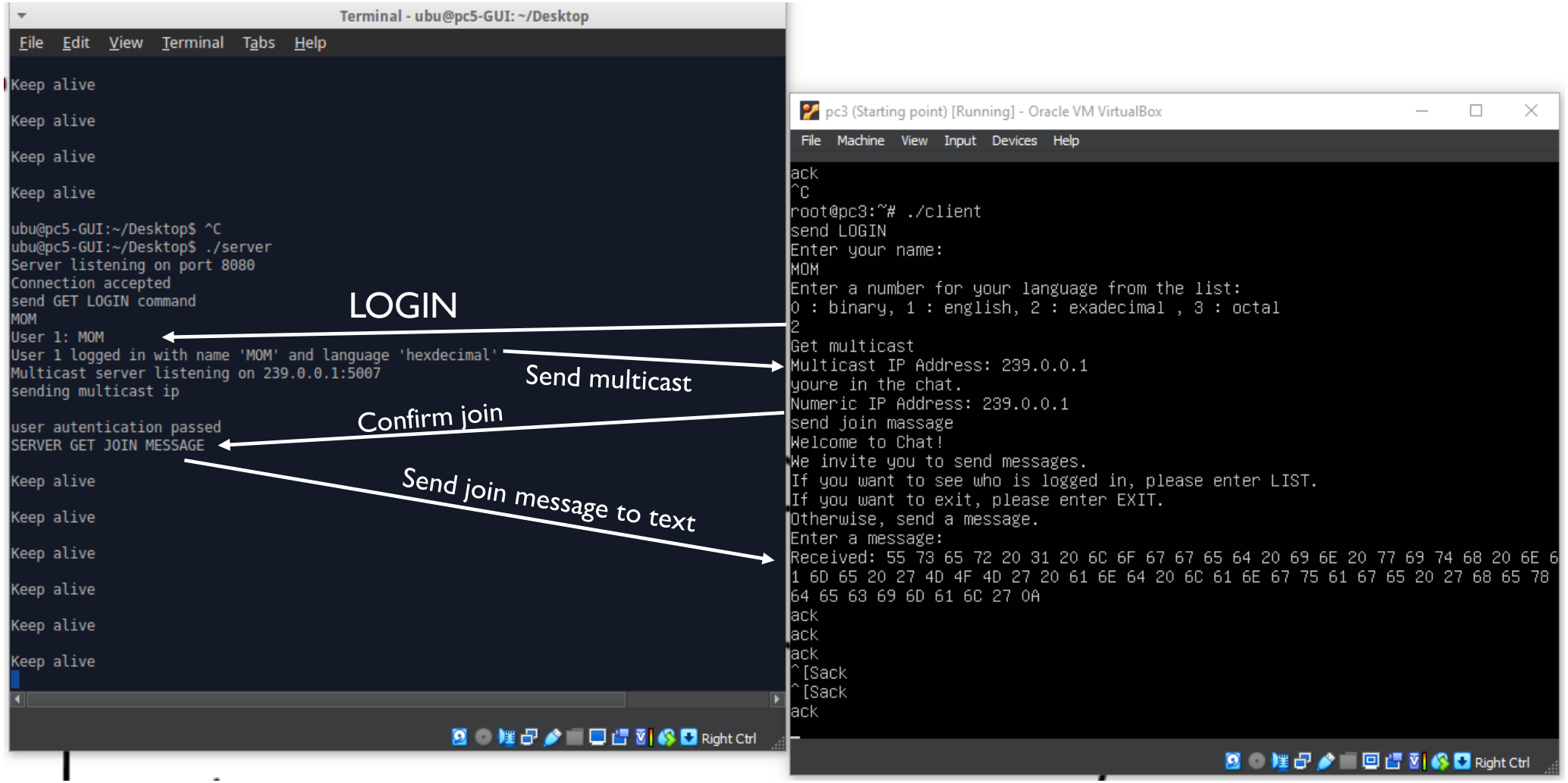




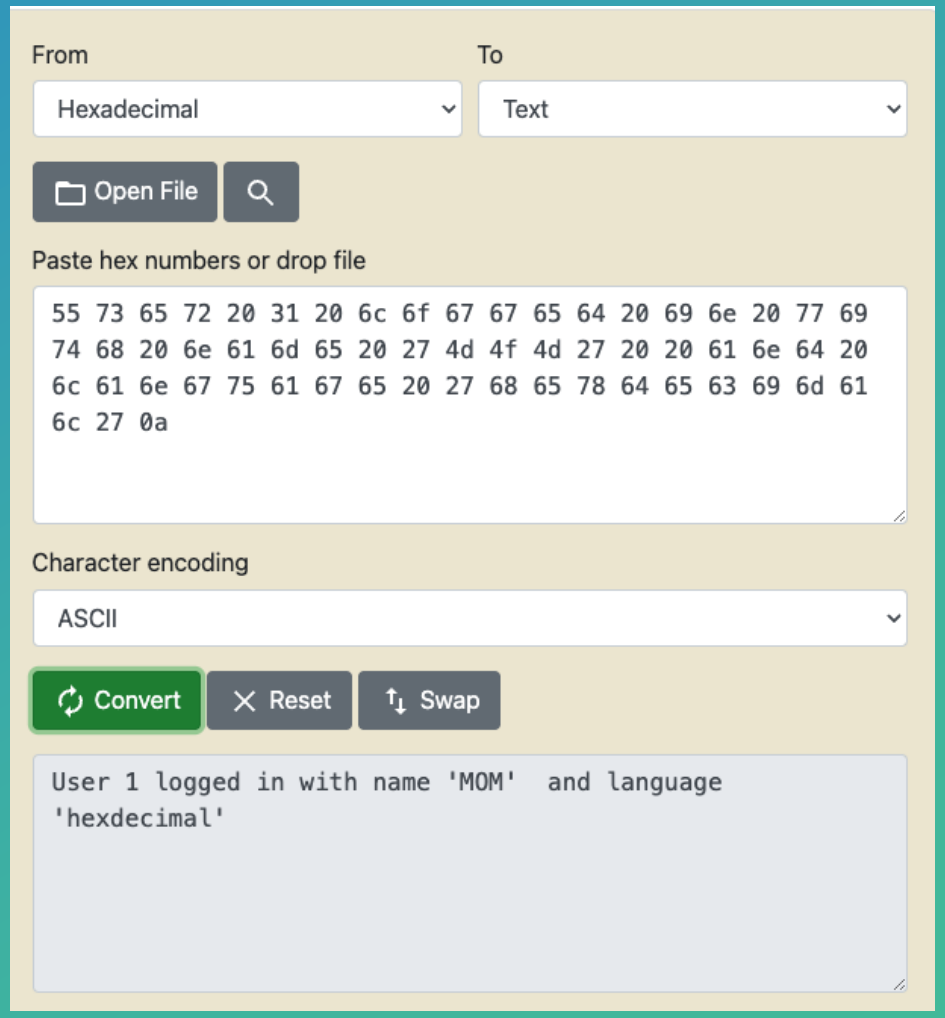
3
CLIENT CHAT


No. Time Source Destination Protocol Length Info
4 10:01:31.573095 192.6.3.254 224.0.0.13 PIMv2 130 Bootstrap
5 10:01:53.418672 192.6.1.1 192.6.3.1 TCP 74 47921 → 8080 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SAC
6 10:01:53.419647 192.6.3.1 192.6.1.1 TCP 74 8080 → 47921 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 M
7 10:01:53.461636 192.6.1.1 192.6.3.1 TCP 66 47921 → 8080 [ACK] Seq=1 Ack=1 Win=29216 Len=0 TSval=
8 10:01:53.462610 192.6.3.1 192.6.1.1 TCP 1082 8080 → 47921 [PSH, ACK] Seq=1 Ack=1 Win=29056 Len=101
9 10:01:53.493759 192.6.1.1 192.6.3.1 TCP 66 47921 → 8080 [ACK] Seq=1 Ack=1017 Win=32096 Len=0 TSv
10 10:01:57.260750 192.6.1.1 192.6.3.1 TCP 1082 47921 → 8080 [PSH, ACK] Seq=1 Ack=1017 Win=32096 Len=
11 10:01:57.260750 192.6.3.1 192.6.1.1 TCP 66 8080 → 47921 [ACK] Seq=1017 Ack=1017 Win=31872 Len=0
12 10:01:59.260628 192.6.3.1 192.6.1.1 TCP 1082 8080 → 47921 [PSH, ACK] Seq=1017 Ack=1017 Win=31872 L
13 10:01:59.283018 192.6.1.1 192.6.3.1 TCP 66 47921 → 8080 [ACK] Seq=1017 Ack=2033 Win=35008 Len=0
14 10:01:59.293725 192.6.1.1 192.6.3.1 TCP 1082 47921 → 8080 [PSH, ACK] Seq=1017 Ack=2033 Win=35008 L
Frame 9: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface -, id 0
Ethernet II, Src: c4:04:28:50:00:00 (c4:04:28:50:00:00), Dst: PcsCompu_42:e2:35 (08:00:27:42:e2:35)
Internet Protocol Version 4, Src: 192.6.1.1, Dst: 192.6.3.1
Transmission Control Protocol, Src Port: 47921, Dst Port: 8080, Seq: 1, Ack: 1017, Len: 0
Source Port: 47921
Destination Port: 8080
[Stream index: 0]
[TCP Segment Len: 0]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 234616143
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1017 (relative ack number)
Acknowledgment number (raw): 3119121050
0000 08 00 27 42 e2 35 c4 04 28 50 00 00 08 00 45 00
0010 00 34 4f c7 40 00 3e 06 68 ee c0 06 01 01 c0 06
0020 03 01 bb 31 1f 90 0d fb f5 4f b9 ea 02 9a 80 10
0030 03 eb bd 9a 00 00 01 01 08 0a 00 00 ba 9e 00 02
0040 db f6
WIRESHARK OBSERVATIONS
LOGIN COMMAND

ESTABLISH


Terminal - ubu@pc5-GUI: ~/Desktop
File Edit View Terminal Tabs Help
Keep alive
Keep alive
Keep alive
Keep alive
ubu@pc5-GUI:~/Desktop$ ^C
ubu@pc5-GUI:~/Desktop$ ./server
Server listening on port 8080
Connection accepted
send GET LOGIN command
MOM
User 1: MOM
User 1 logged in with name 'MOM' and language 'hexdecimal'
Multicast server listening on 239.0.0.1:5007
sending multicast ip
user autentication passed
SERVER GET JOIN MESSAGE
Keep alive
Keep alive
Keep alive
Keep alive
Keep alive
Keep alive
Keep alive
Right Ctrl
LOGIN
Send multicast
Confirm join
Send join message to text
pc3 (Starting point) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
ack
^C
root@pc3:~# ./client
send LOGIN
Enter your name:
MOM
Enter a number for your language from the list:
0 : binary, 1 : english, 2 : exadecimal , 3 : octal
2
Get multicast
Multicast IP Address: 239.0.0.1
youre in the chat.
Numeric IP Address: 239.0.0.1
send join massage
Welcome to Chat!
We invite you to send messages.
If you want to see who is logged in, please enter LIST.
If you want to exit, please enter EXIT.
Otherwise, send a message.
Enter a message:
Received: 55 73 65 72 20 31 20 6C 6F 67 67 65 64 20 69 6E 20 77 69 74 68 20 6E 6
1 6D 65 20 27 4D 4F 4D 27 20 61 6E 64 20 6C 61 6E 67 75 61 67 65 20 27 68 65 78
64 65 63 69 6D 61 6C 27 0A
ack
ack
ack
^[Sack
^[Sack
ack
Right Ctrl

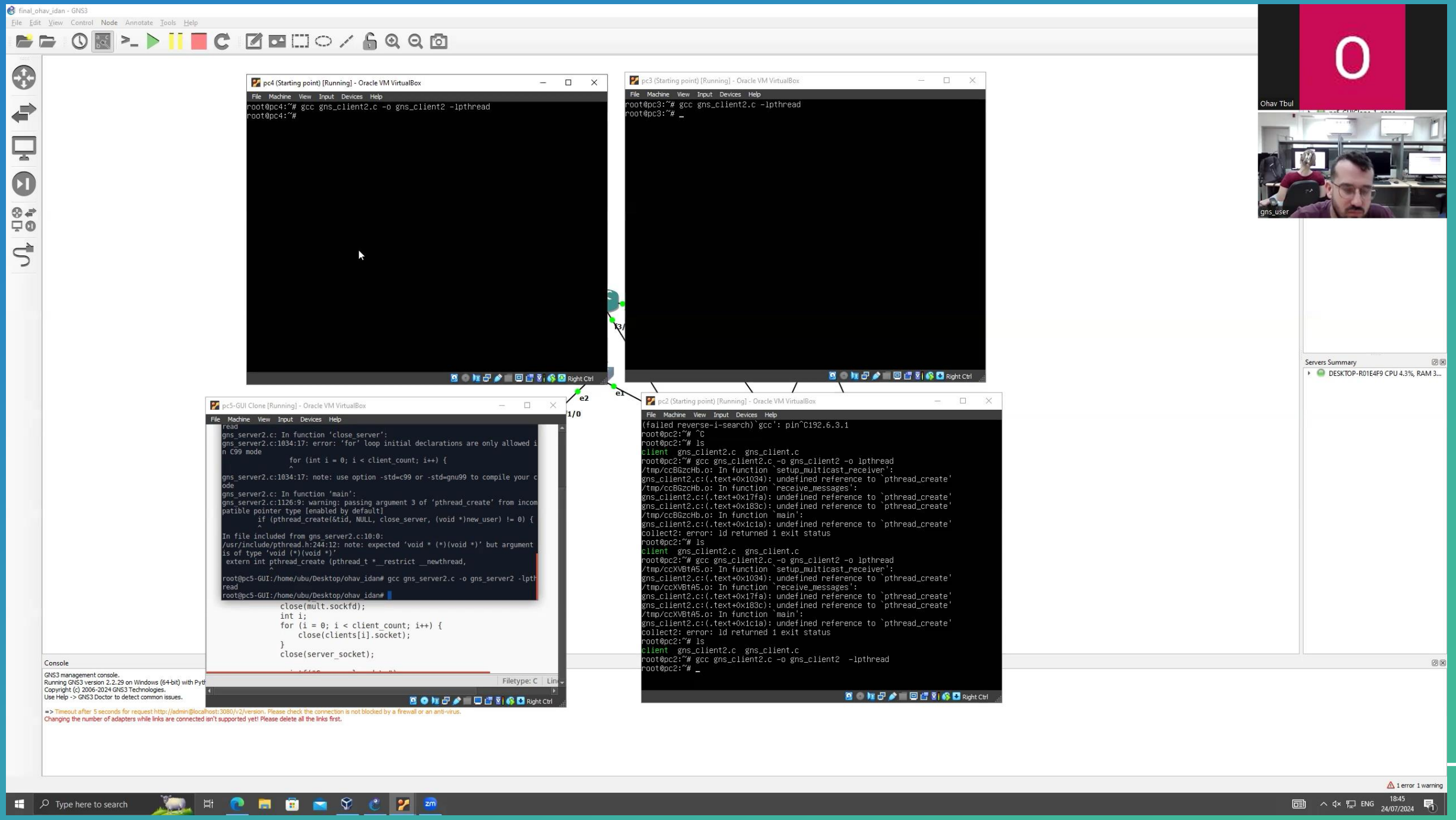
WE SAVED YOU TIME ALON!



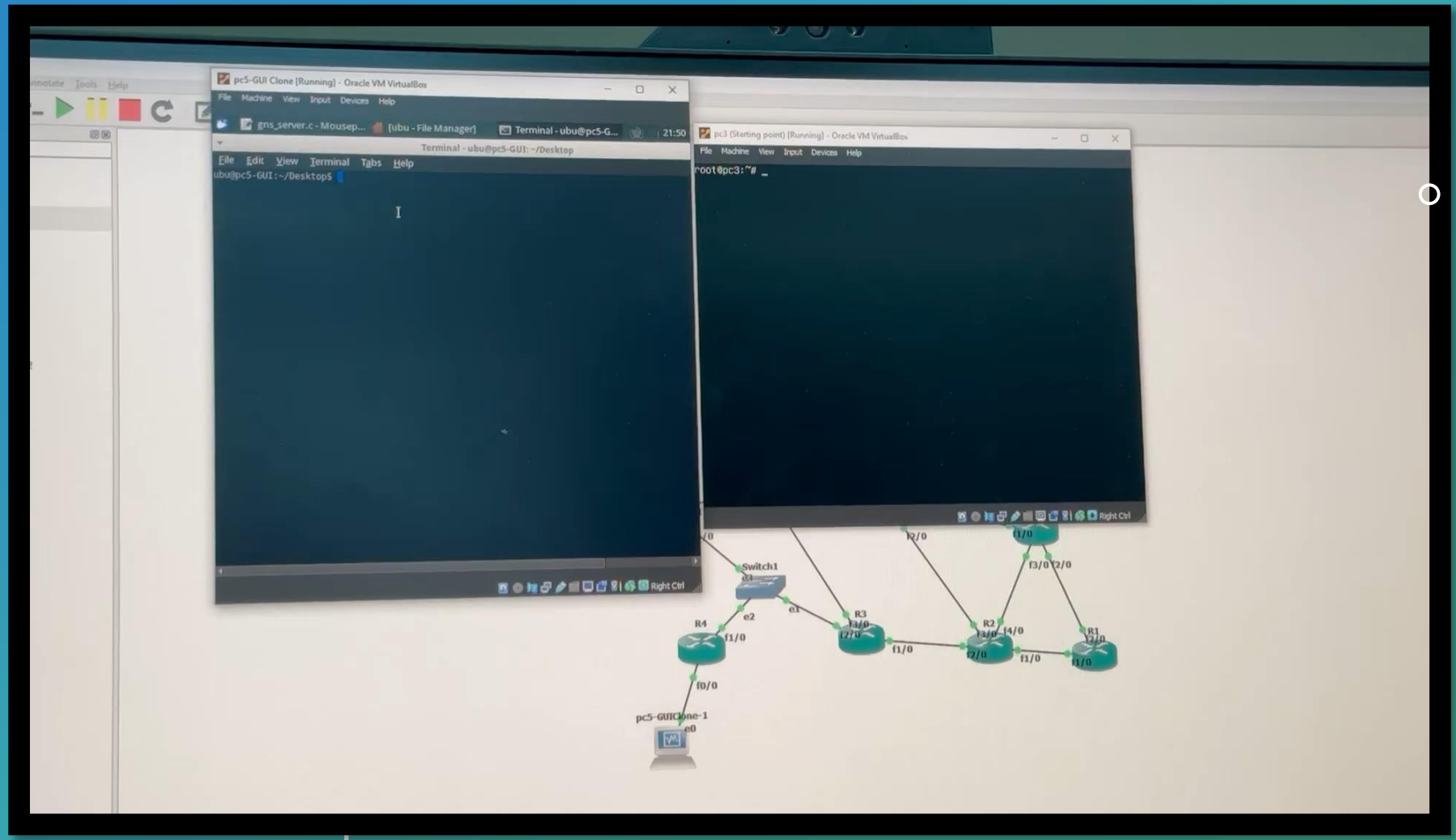
CONVERSATION

OBSERVATIONS

- In the end of the videos the server continue working even if there is no clients in the chat
- chat conversation
- List of update users
- Establish process
- Exit and enter multicast messages


final_ohav_idan - GNS3
File Edit View Control Node Annotate Tools Help
pc4 (Starting point) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@pc4:~# gcc gns_client2.c -o gns_client2 -lpthread
root@pc4:~#
Right Ctrl
pc3 (Starting point) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
root@pc3:~# gcc gns_client2.c -lpthread
root@pc3:~#
Right Ctrl
pc5-GUI Clone [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
gns_server2.c: In function 'close_server':
gns_server2.c:1034:17: error: 'for' loop initial declarations are only allowed in C99 mode
for (int i = 0; i < client_count; i++) {
gns_server2.c:1034:17: note: use option -std=c99 or -std=gnu99 to compile your code
gns_server2.c: In function 'main':
gns_server2.c:1126:9: warning: passing argument 3 of 'pthread_create' from incompatible pointer type [enabled by default]
if (pthread_create(&tid, NULL, close_server, (void *)new_user) != 0) {
In file included from gns_server2.c:10:0:
/usr/include/pthread.h:244:12: note: expected 'void * (*)(void *)' but argument is of type 'void (*)(void *)'
extern int pthread_create (pthread_t *__restrict __newthread,
root@pc5-GUI:/home/ubu/Desktop/ohav_idan# gcc gns_server2.c -o gns_server2 -lpthread
root@pc5-GUI:/home/ubu/Desktop/ohav_idan#
close(mult.sockfd);
int i;
for (i = 0; i < client_count; i++) {
close(clients[i].socket);
}
close(server_socket);
Filetype: C
Right Ctrl
pc2 (Starting point) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
(failed reverse-i-search)`gcc': pin^C192.6.3.1
root@pc2:~# ^C
root@pc2:~# ls
client gns_client2.c gns_client.c
root@pc2:~# gcc gns_client2.c -o gns_client2 -o lpthread
/tmp/ccBGzcHb.o: In function `setup_multicast_receiver':
gns_client2.c:(.text+0x1034): undefined reference to `pthread_create'
/tmp/ccBGzcHb.o: In function `receive_messages':
gns_client2.c:(.text+0x17fa): undefined reference to `pthread_create'
gns_client2.c:(.text+0x183c): undefined reference to `pthread_create'
/tmp/ccBGzcHb.o: In function `main':
gns_client2.c:(.text+0x1c1a): undefined reference to `pthread_create'
collect2: error: ld returned 1 exit status
root@pc2:~# ls
client gns_client2.c gns_client.c
root@pc2:~# gcc gns_client2.c -o gns_client2 -o lpthread
/tmp/ccXVBtA5.o: In function `setup_multicast_receiver':
gns_client2.c:(.text+0x1034): undefined reference to `pthread_create'
/tmp/ccXVBtA5.o: In function `receive_messages':
gns_client2.c:(.text+0x17fa): undefined reference to `pthread_create'
gns_client2.c:(.text+0x183c): undefined reference to `pthread_create'
/tmp/ccXVBtA5.o: In function `main':
gns_client2.c:(.text+0x1c1a): undefined reference to `pthread_create'
collect2: error: ld returned 1 exit status
root@pc2:~# ls
client gns_client2.c gns_client.c
root@pc2:~# gcc gns_client2.c -o gns_client2 -lpthread
root@pc2:~#
Right Ctrl
e2
e1
1/0
Ohav Tbul
gns_user
Servers Summary
DESKTOP-R01E4F9 CPU 4.3%, RAM 3...
Console
GNS3 management console.
Running GNS3 version 2.2.29 on Windows (64-bit) with Pyth
Copyright (c) 2006-2024 GNS3 Technologies.
Use Help -> GNS3 Doctor to detect common issues.
=> Timeout after 5 seconds for request http://admin@localhost:3080/v2/version. Please check the connection is not blocked by a firewall or an anti-virus.
Changing the number of adapters while links are connected isn't supported yet! Please delete all the links first.
1 error 1 warning
Type here to search
ENG
18:45
24/07/2024

CLIENT STOP WORKING IN ESTABLISH



THANK YOU