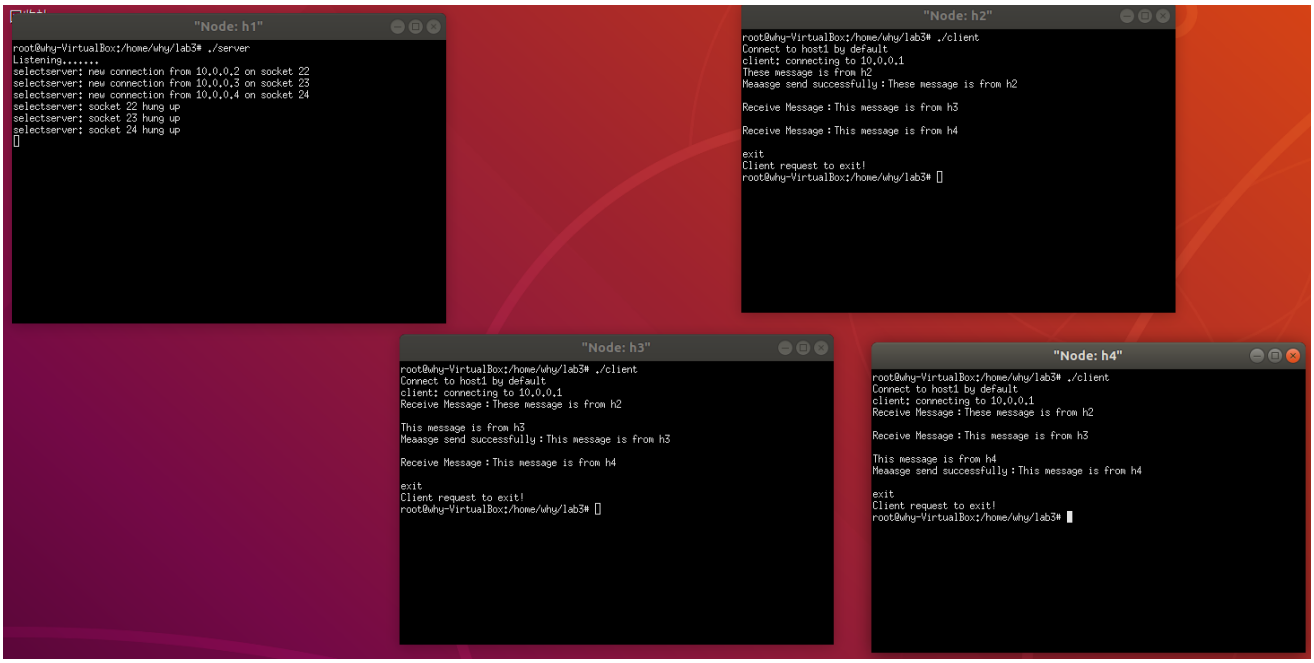
1. Question 1

   I fix the host 1 as the server and others as client. So I hardcode the host 1 as server in the client.cpp. If you want to change the server, please change the fixed server address in the client.cpp.

   According to the screenshots, we can see that the server can know which socket port is used for a client and the IP addresses of clients. The clients can send and receive the messages successfully.

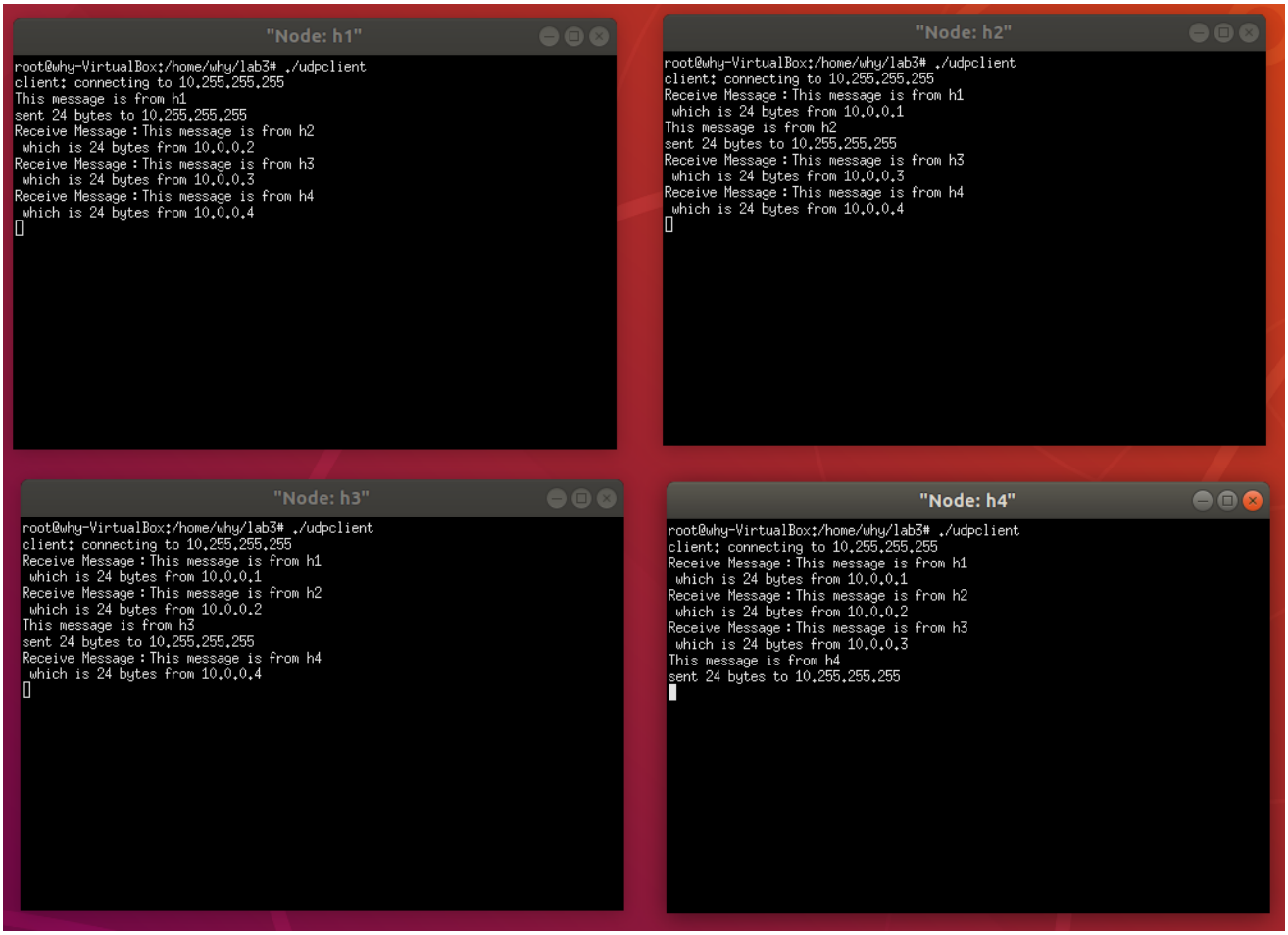
   For example, if the host 2 sends a message "This message is from h2", then the host 3 and 4 will receive this message exactly.



2. Question 2 and Bonus question

   I implement the client-only model with UDP protocol and realize the function that each client's terminal won't display the message sent by the client itself. The results are as follows:

   For example, if host 1 sends a message "This message is from h1", then host 2, 3 and 4 will all receive this message while host 1 won't receive this message.



The main difficulty of Bonus question is to find the client own IP address. If we have known the client own IP address, we just need to check if the IP address of the received message is the same as the client own IP address. These codes is used to get the client own IP address:

```
int local_sockfd;
struct ifconf ifconf;
struct ifreq *ifreq;
char tmp_buf[1024];
char* local_ip;
//初始化ifconf
ifconf.ifc_len = 1024;
ifconf.ifc_buf = tmp_buf;

if((local_sockfd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
{
    perror("socket error");
    exit(1);
}

//获取所有接口信息
ioctl(local_sockfd, SIOCGIFCONF, &ifconf);

//逐个获取Ip地址
ifreq = (struct ifreq*)tmp_buf;
for(int i = (ifconf.ifc_len/sizeof(struct ifreq)); i>0; i--)
{
    // printf("name = [%s] : ",ifreq->ifr_name);
    local_ip=inet_ntoa( ((struct sockaddr_in *)&(ifreq->ifr_addr))->sin_addr);
    ifreq++;
}
strcpy(local_addr, local_ip);//get local IP address
close(local_sockfd);
```