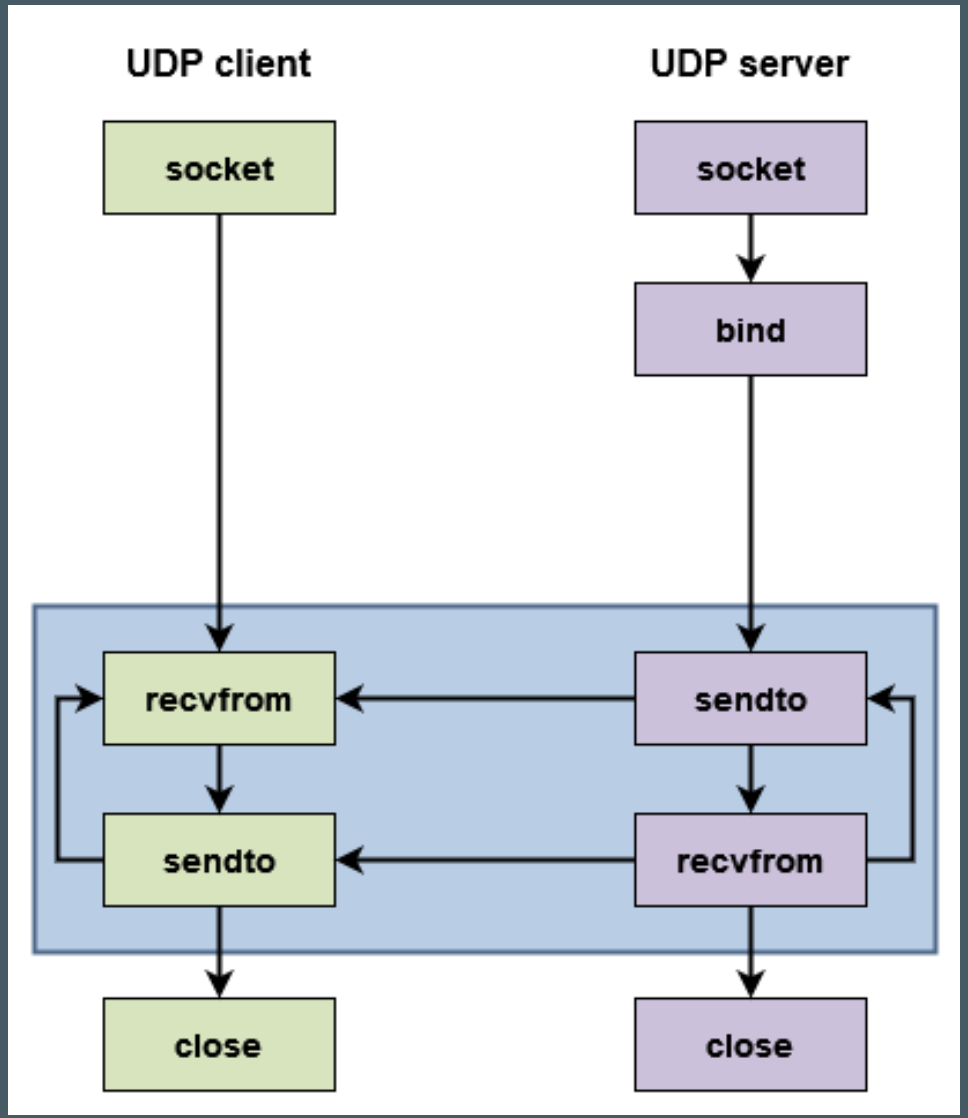
Telecommunications Network

Practice 5

UDP


UDP client
UDP server
socket
socket
bind
recvfrom
sendto
sendto
recvfrom
close
close

UDP

- socket

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

- recvfrom()

```
data, address = sock.recvfrom(4096)
```

- sendto()

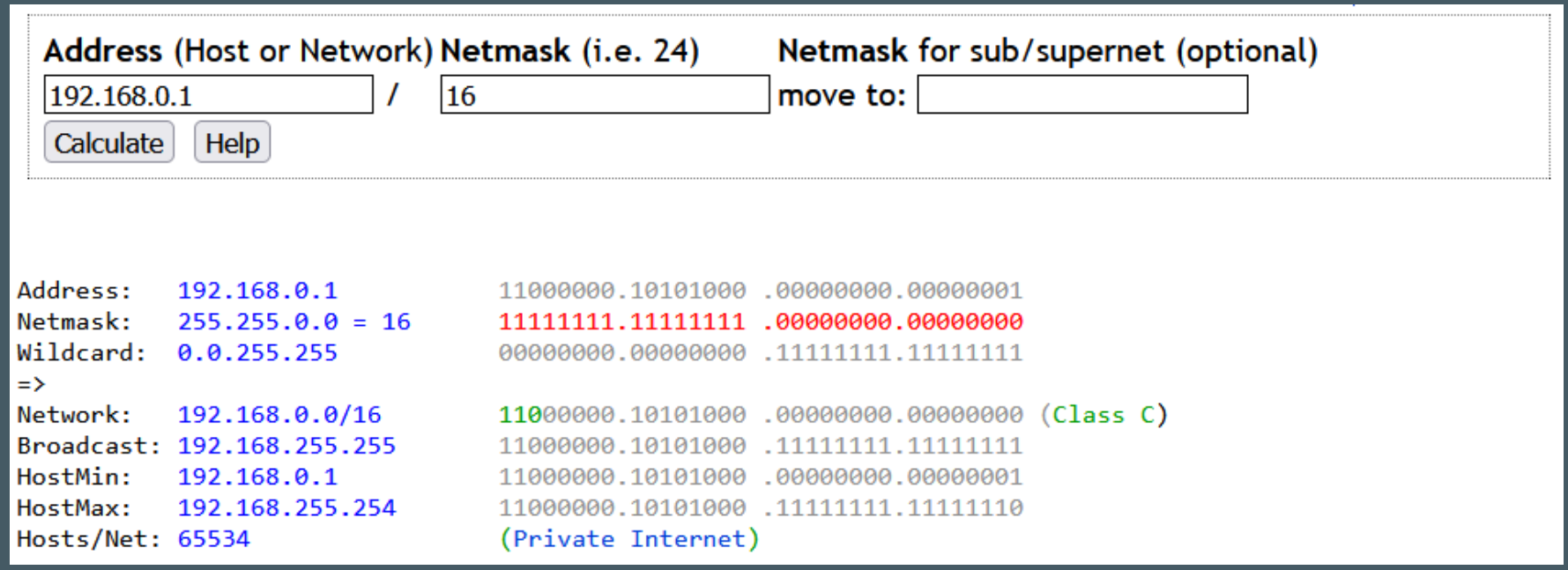
```
sent = sock.sendto(data, address)
```

Exercise I.

- Write a server-client application that uses UDP.
- The client should send a b”Hello Server” bytestring to the server and in return the server should respond with b”Hello Client”.
- Is our server capable of handling multiple clients? Why or why not?

Netmask

- Description of addresses in a subnet.



Netmask RFC

CIDR RFC

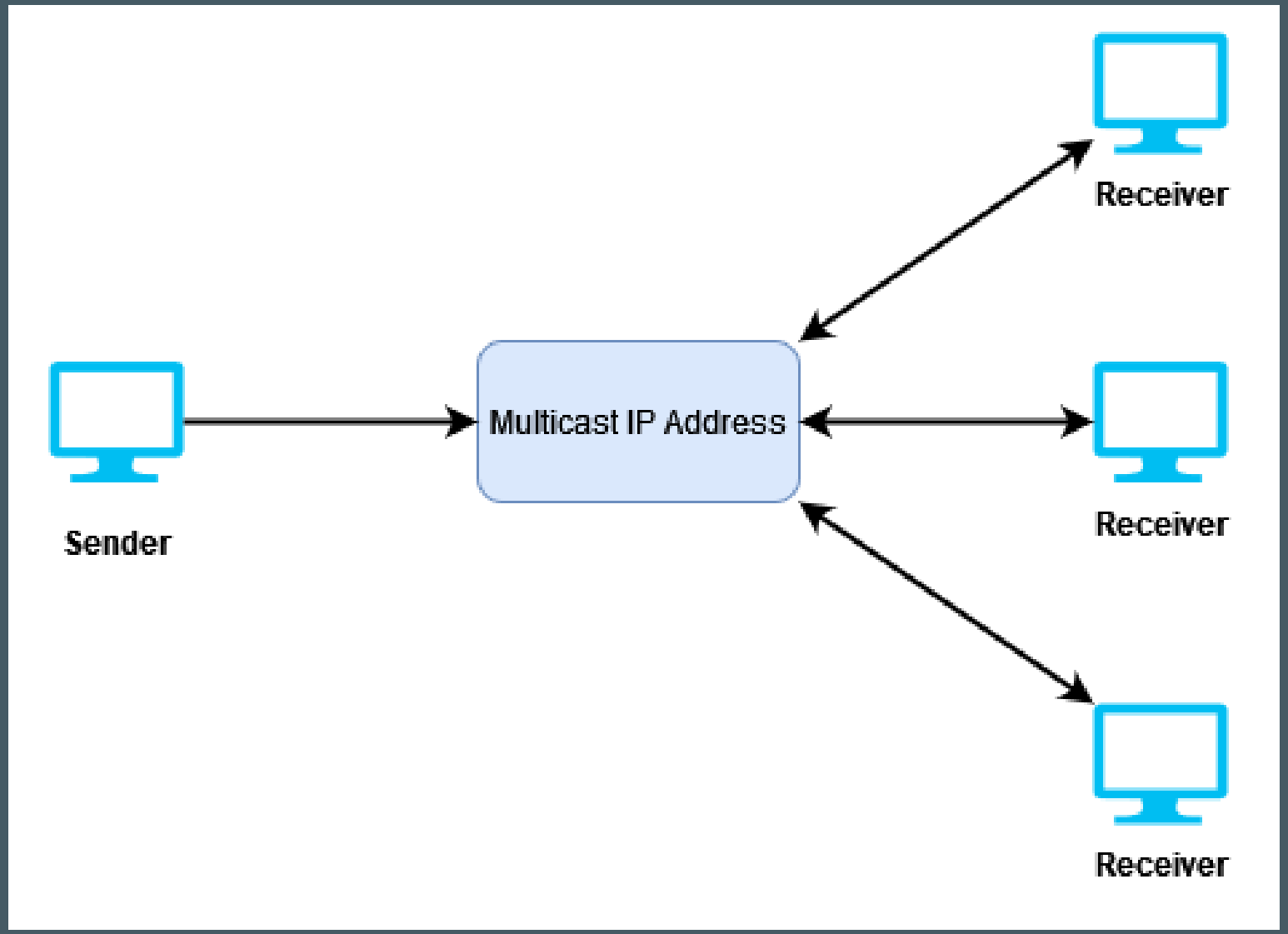
Exercise II.

- How many addresses do we get with the following masks?
- Calculate the min and max address for each option.
  - 188.100.22.12/32
  - 188.100.22.12/20
  - 188.100.22.12/10

Multicast

Class	Range	Description
A	0.0.0.0 - 127.255.255.255	Unicast
B	128.0.0.0 - 191.255.255.255	Unicast
C	192.0.0.0 - 223.255.255.255	Unicast
D	224.0.0.0 - 239.255.255.255	Multicast
E	240.0.0.0 - 255.255.255.255	Reserved

Multicast



Multicast

- setsockopt() (sender)

```
ttl = struct.pack("b", 1)
sock.setsockopt(socket.IPPROTO_IP, socket.IP_MULTICAST_TTL, ttl)
```

- Adding a socket to the multicast group (recv)

```
multicast_group = "224.3.29.71"
group = socket.inet_aton(multicast_group)
mreq = struct.pack("4sL", group, socket.INADDR_ANY)
sock.setsockopt(socket.IPPROTO_IP, socket.IP_ADD_MEMBERSHIP, mreq)
```

Udp stream example

- Example code on the website.
- Install OpenCV

```
python3 –m pip install --user opencv-python
```

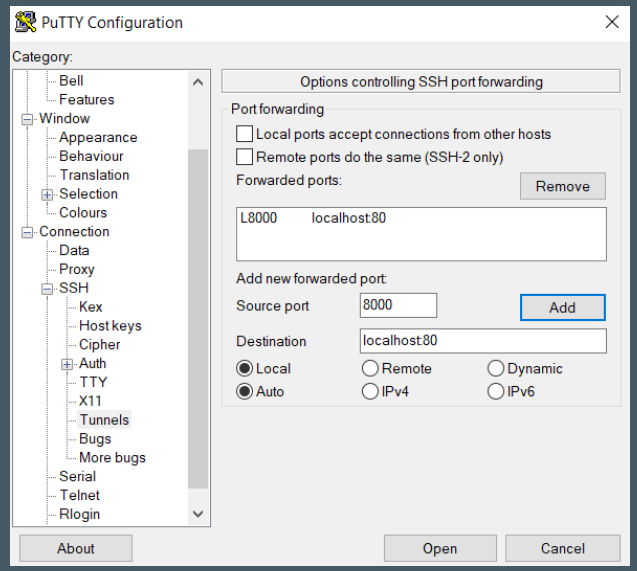
How video streaming works

SSH Tunnel

- In the terminal (Works on Windows as well!):

```
ssh –L 8000:localhost:80 user@hostname
```

- Another option can be the use of a friendlier (one with a GUI) ssh client e.g. Putty.



Exercise III.

- Modify the calculator application so that it uses UDP instead of TCP.

Exercise IV.

- Write a server-client application, where the client send a picture to the server over UDP.
  - The client should send the file in 200 byte chunks.
  - If it gets to the end of the file, the client should send an empty string.
  - The server should acknowledge every chunk it receives with the b”OK” bytestring.