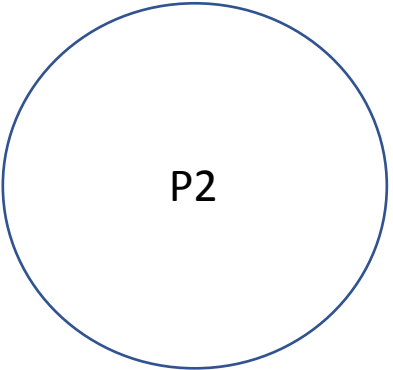
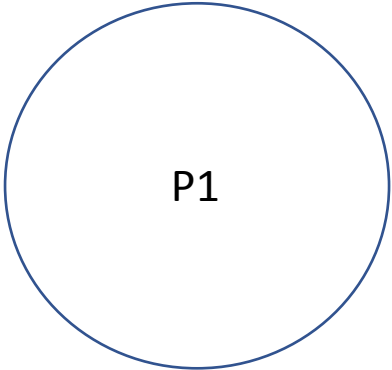
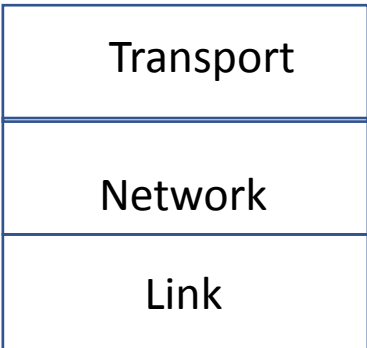
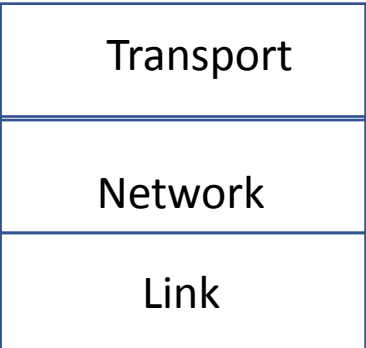


Sockets

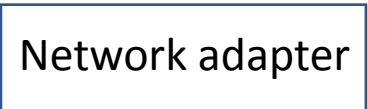
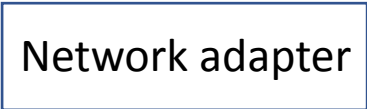


User space

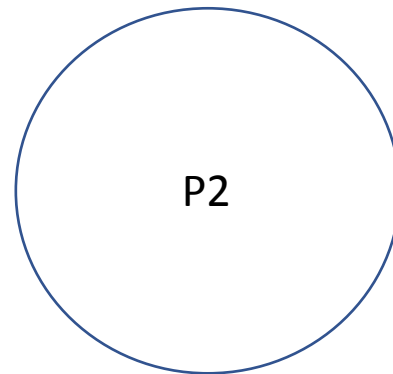
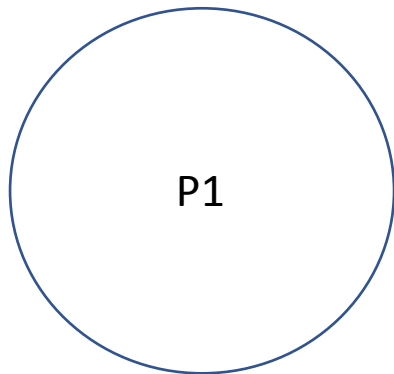
Kernel space



Hardware

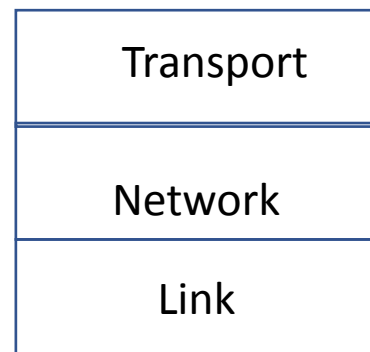
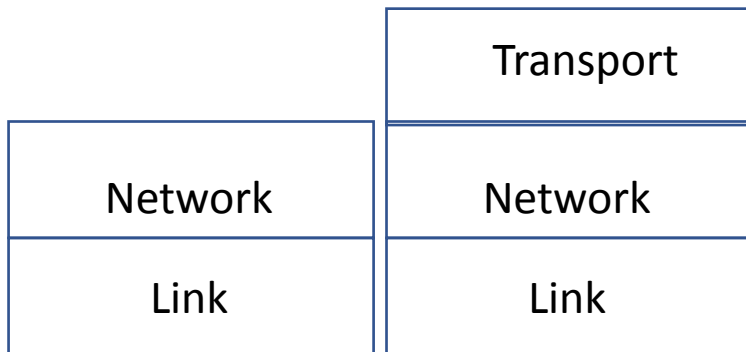
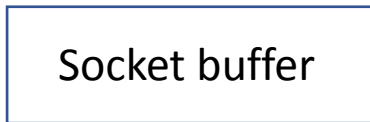


RAW

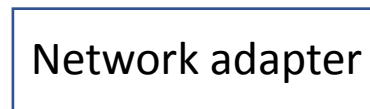
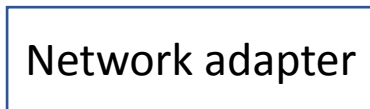


User space

Kernel space



Hardware



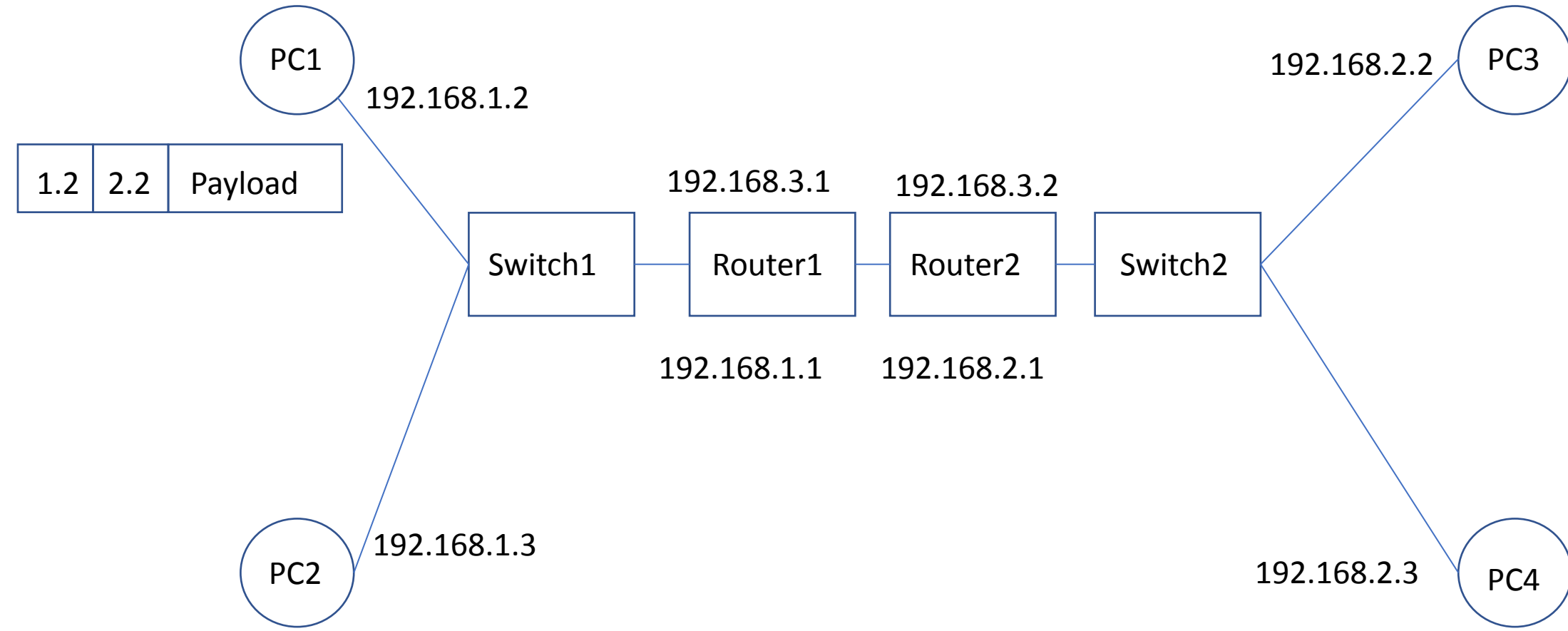
## UDP Header



```
Fd = socket(AF_INET, SOCK_RAW, IPPROTO_UDP);  
Buf -> udpheader + "hello!";  
serv;  
Sendto(buf, serv);  
  
While(1){  
    recvfrom(buf);  
  
}
```

Network	Mask	Gateway	Ethernet if
192.168.1.0	255.255.255.0	0.0.0.0	eth0
0.0.0.0	0.0.0.0	192.168.1.1	eth0

Network	Mask	NextHop	Ethernet if
192.168.1.0	255.255.255.0	0.0.0.0	eth0
192.168.3.0	255.255.255.0	0.0.0.0	eth1
192.168.2.0	255.255.255.0	192.168.3.2	eth1



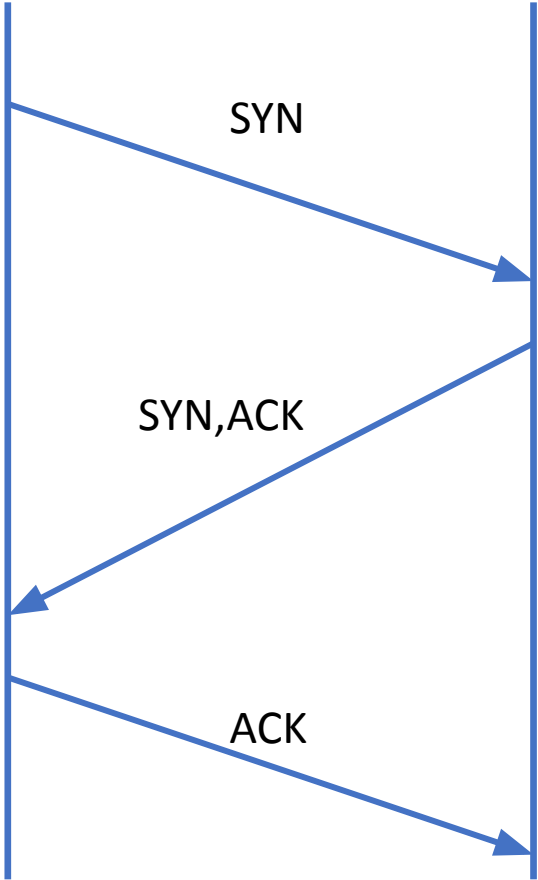
## UDP Header

Source port	Destination port	Length	Check sum	Payload
-------------	------------------	--------	-----------	---------

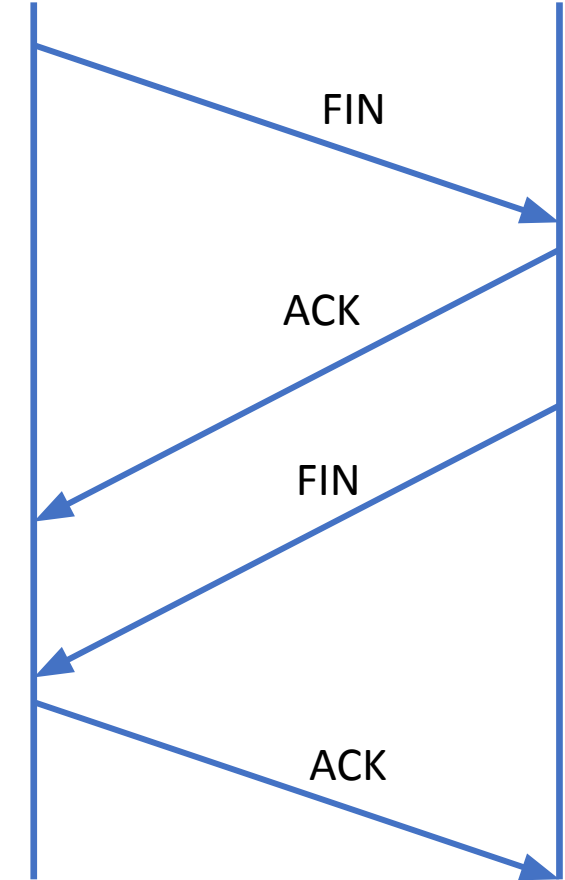
## TCP Header

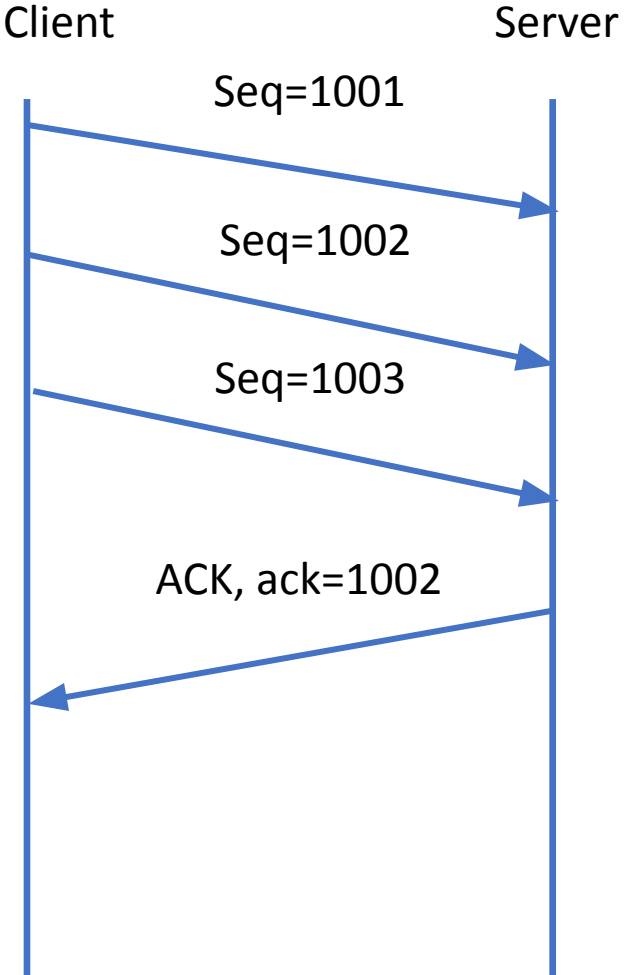
Source port	Destination port	Sequence number	Acknowledgment number	
Offset and flags	Window size	Checksum	Urgent pointer	Options
Payload				

Client Server



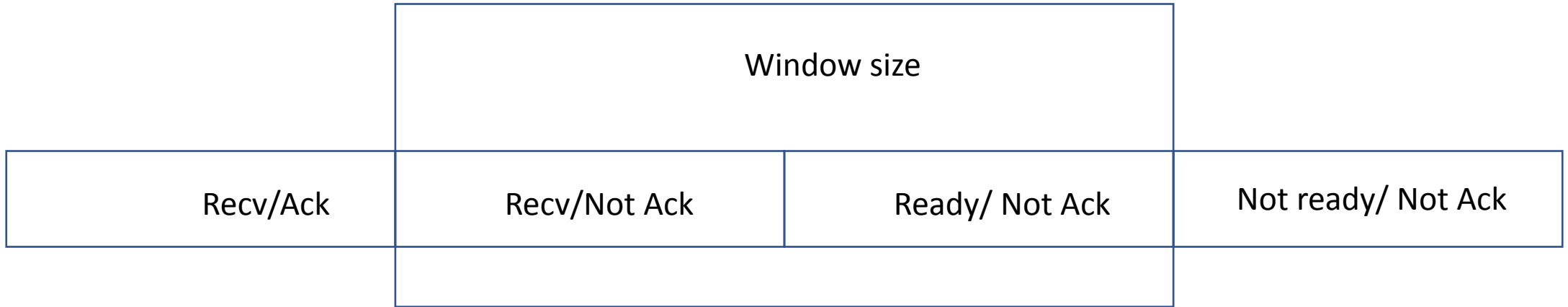
Client Server



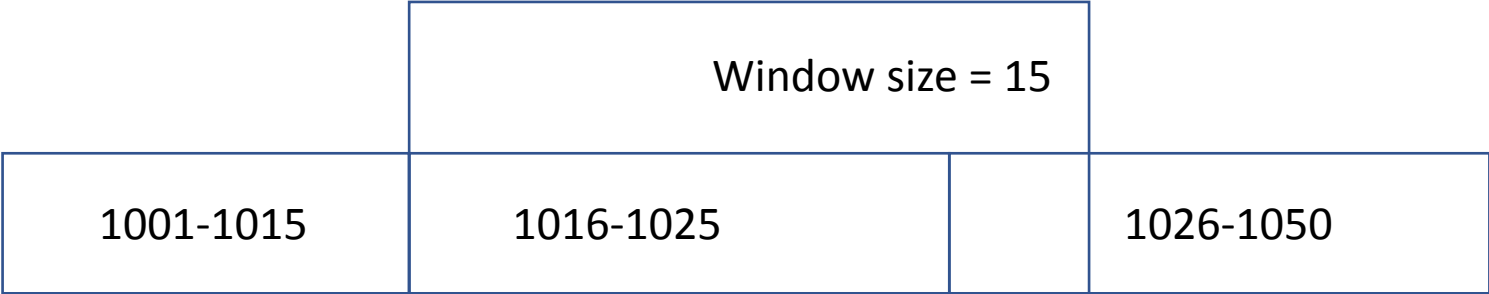


A	B	C
1001	1002	1003

# Window







## IP Header

Ver/IHL	DS	Length	Identification		Flags + Offset
TTL		Transport Proto	Checksum	Source IP	Destination IP
Payload					



## Ethernet Header

lipcap

Destination mac	Source mac	802.1Q	Type	Payload
-----------------	------------	--------	------	---------

```
Fd = socket(AF_PACKET, SOCK_RAW, htons(ETH_P_ALL));
Buf -> macheder + ipheader + udpheader + "hello!";
serv;
Sendto(buf, serv);

While(1){
    recvfrom(buf);
}
```

```
Char buf[128];
Int csum = 0;      001
Short *ptr;      110
                111
Ptr = (short *)buf;
For (i=0;i < 10; i++){
    csum = csum + ptr;
    Ptr++;
}
Tmp = csum >> 16;
Csum = csum + tmp;
Csum = ~cum;
```

## Sockets

### Server(tcp)

```
Struct sockaddr_in serv, client;  
Fd = Socket(AF_INET,SOCK_STREAM,0);  
Bind(serv);  
Listen(fd, 5);  
  
New_fd=accept(fd, client, len);  
Recv(new_fd, buf, size, 0);  
Send(new_fd, buf, size, 0);  
Close(new_fd);  
Close(fd);
```

### Client(tcp)

```
Struct sockaddr_in serv;  
Fd=socket(AF_INET, SOCK_STREAM,0);  
Serv;  
Connect(fd, serv, size);  
Send()  
Recv()  
Close(fd)
```

## Sockets

Server(udp)

Int size;

Struct sockaddr\_in serv, client;

Fd = Socket(AF\_INET,SOCK\_DGRAM,0);

Bind(serv);

Size= sizeof(client);

Recvfrom(fd, buf, size, client, size, 0);

Sendto(fd, buf, size,client, size, 0);

Close(fd);

Client(udp)

Struct sockaddr\_in serv;

Fd=socket(AF\_INET, SOCK\_STREAM,0);

Serv;

Connect(fd, serv, size);

Send(fd, buf, size, 0);

Recv(ds, buf, size, 0);

Close(fd)