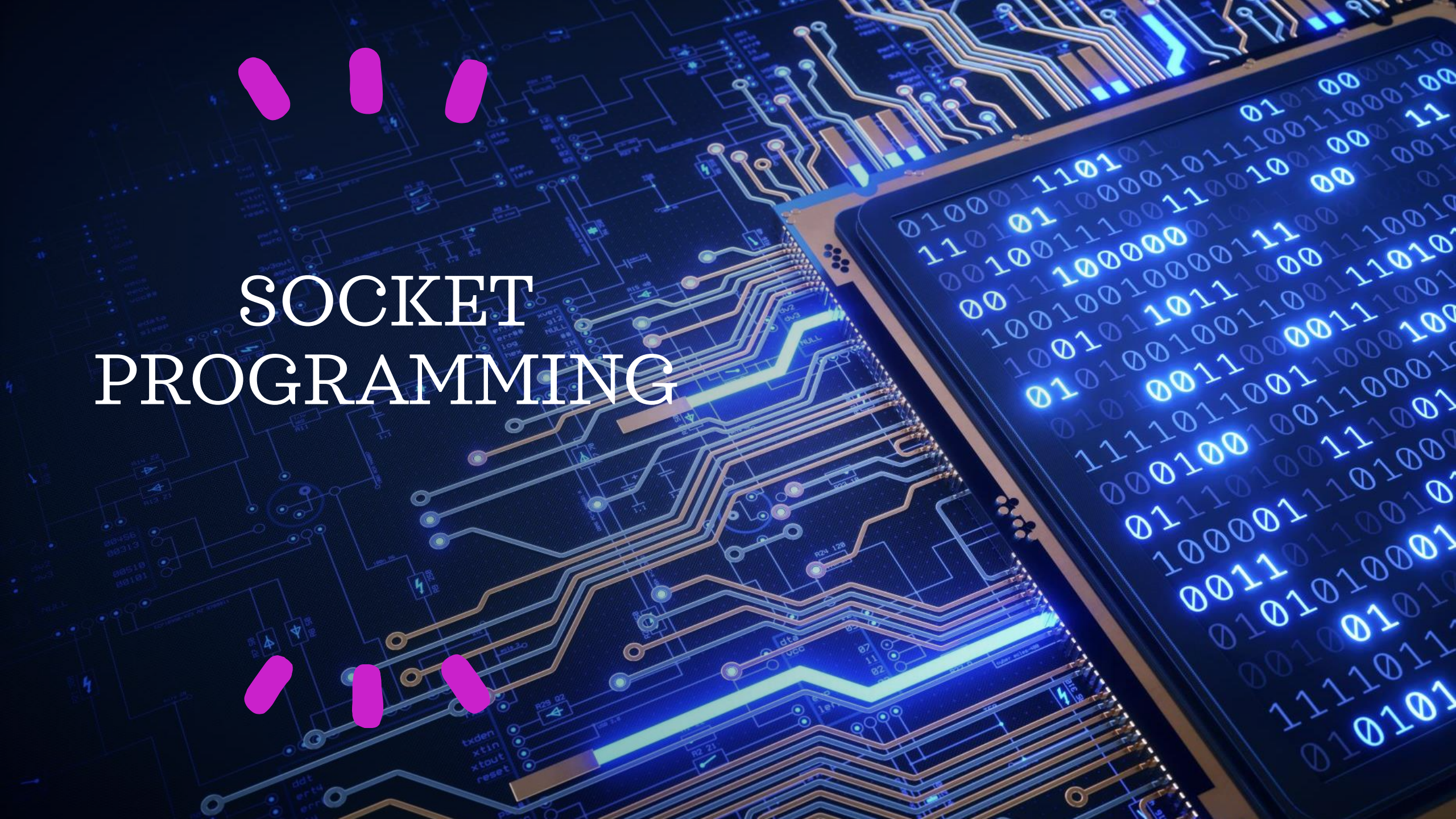
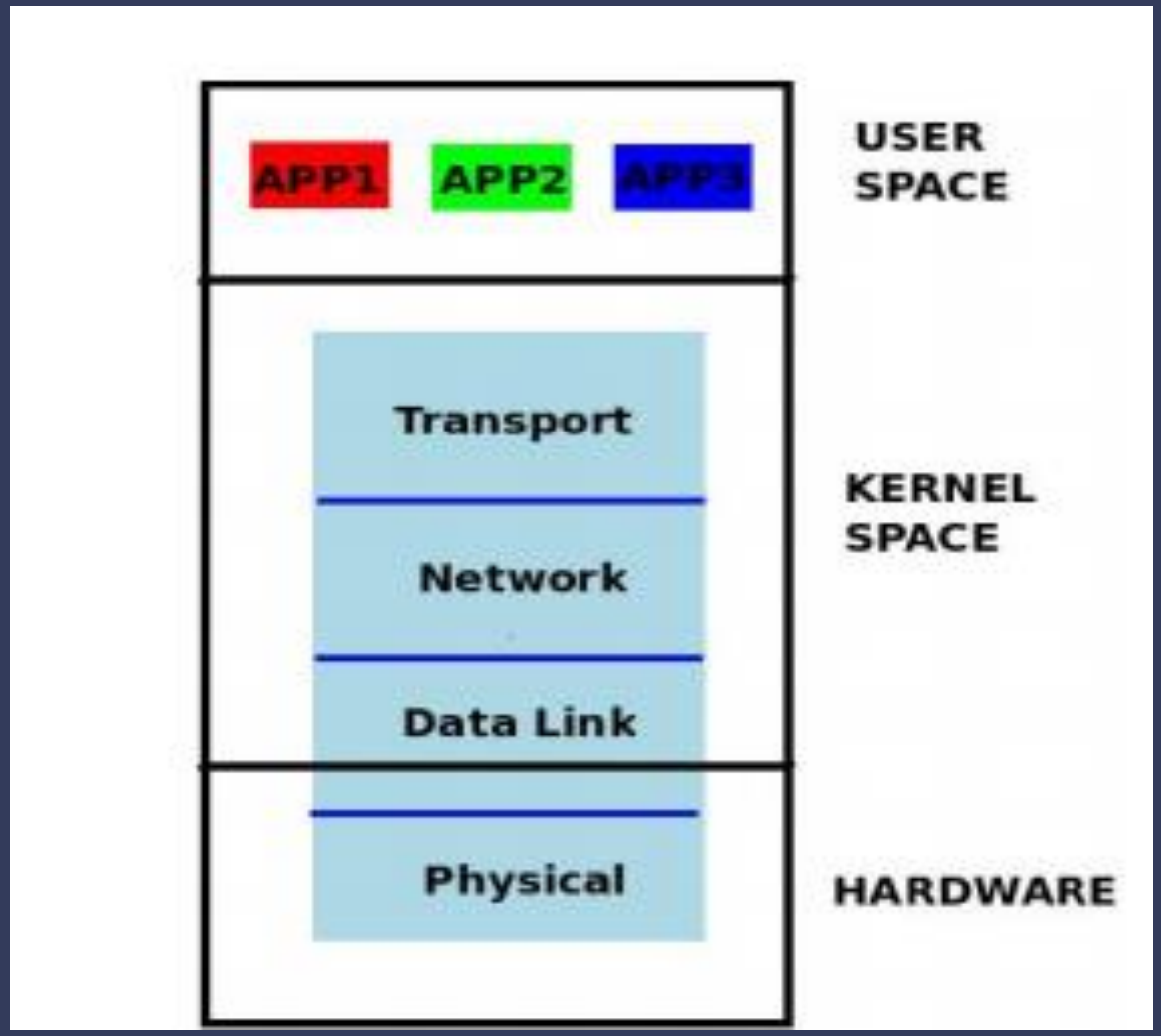
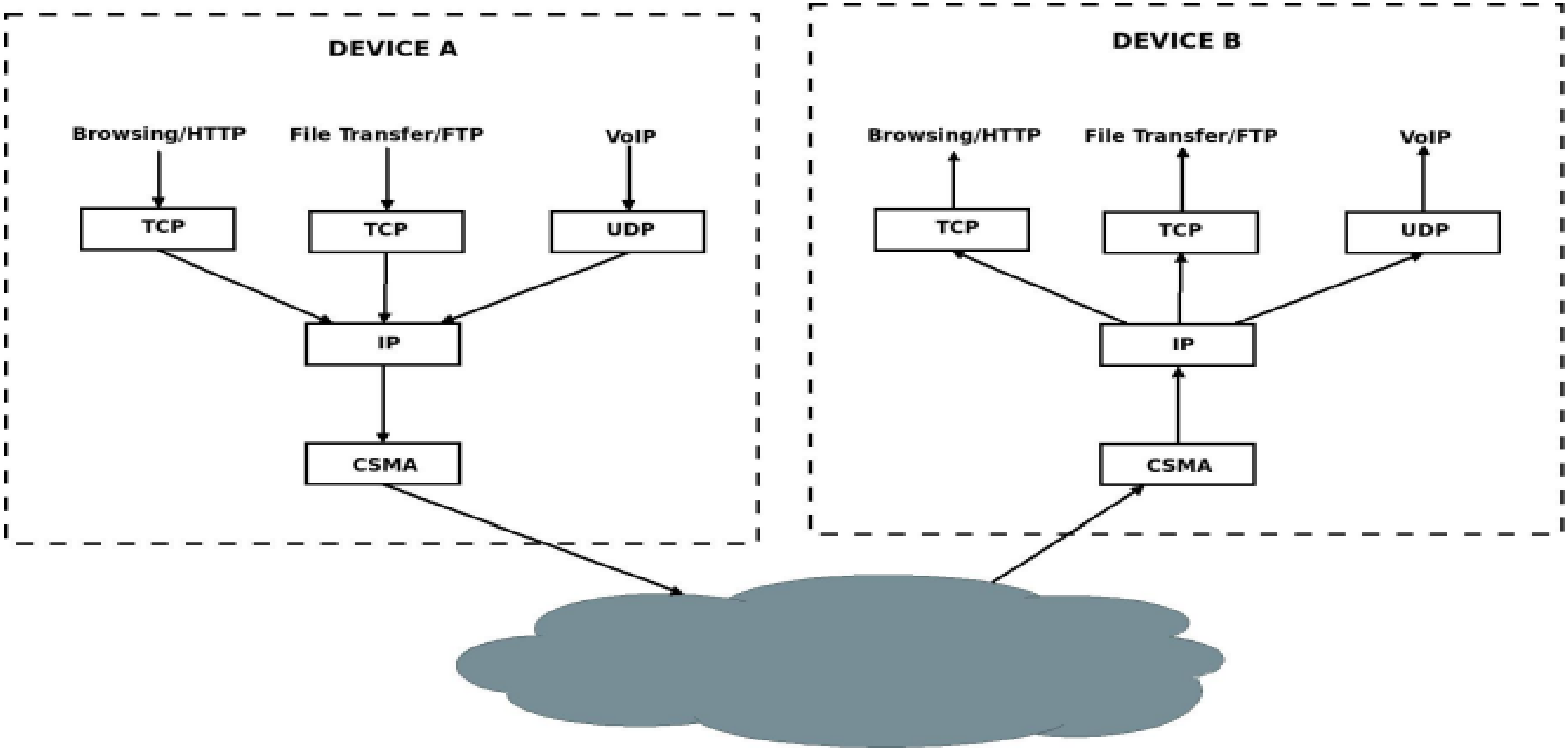

SOCKET
PROGRAMMING

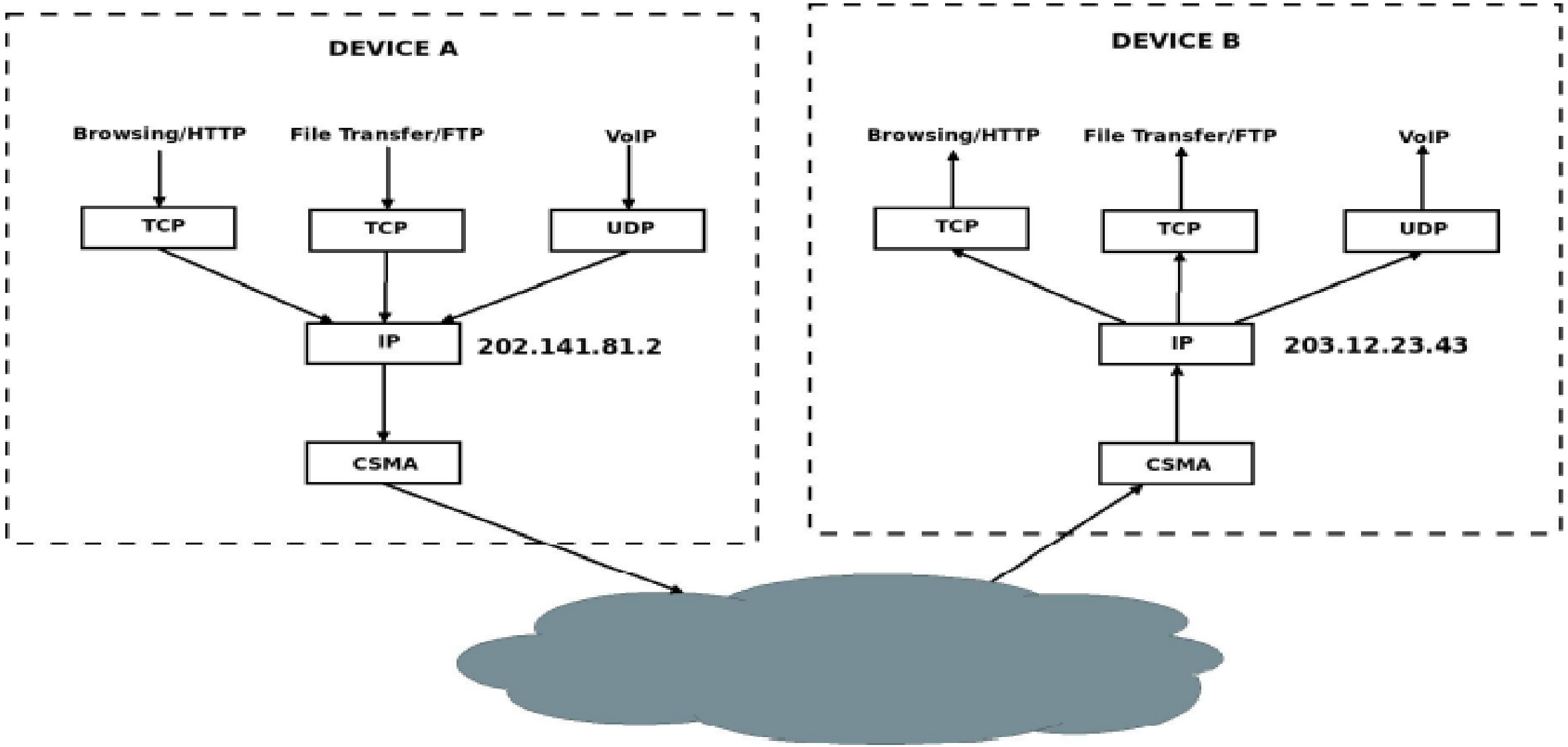
# Integration between network & Operating system



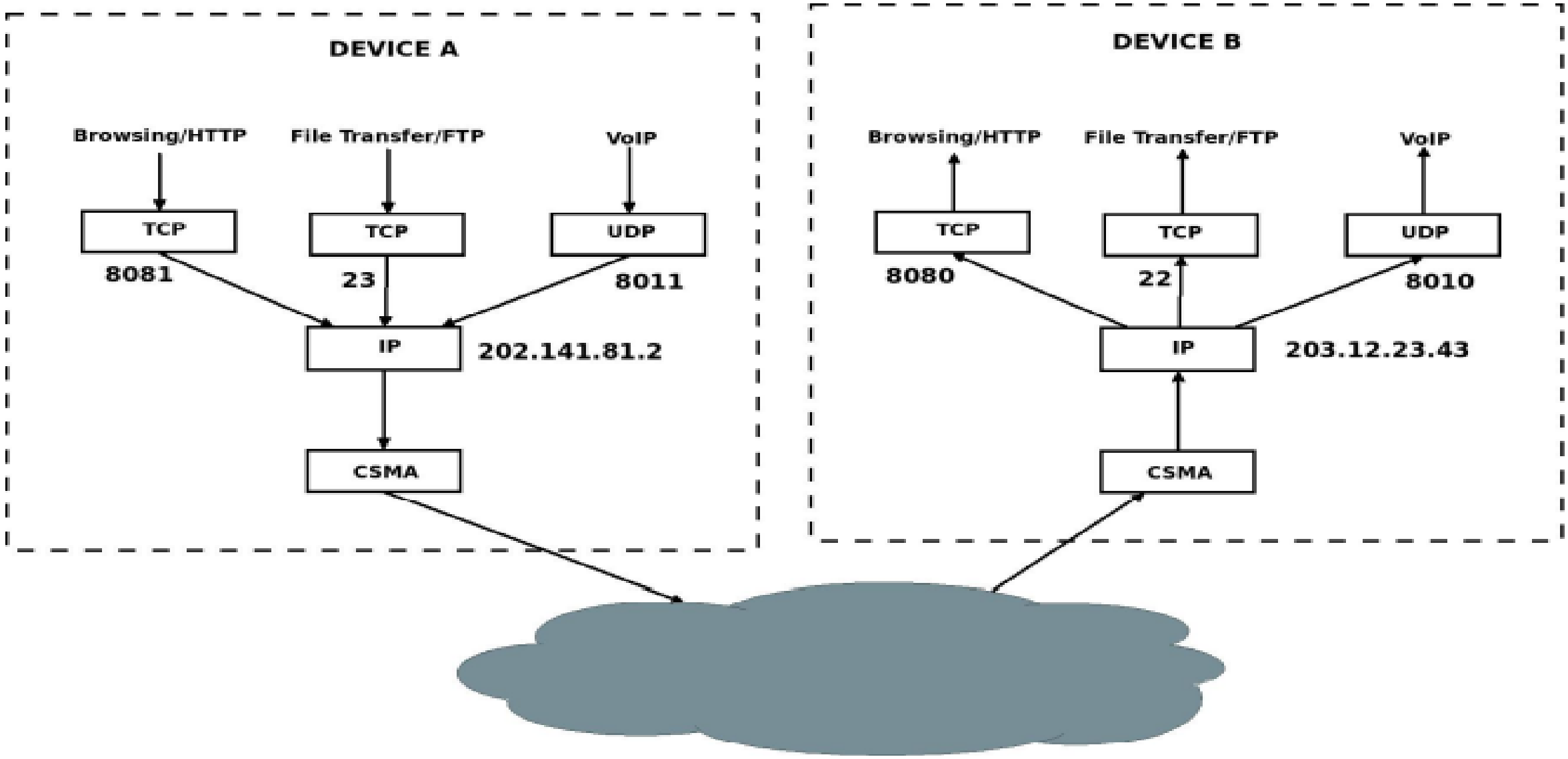
# Application Multiplexing in TCP/IP



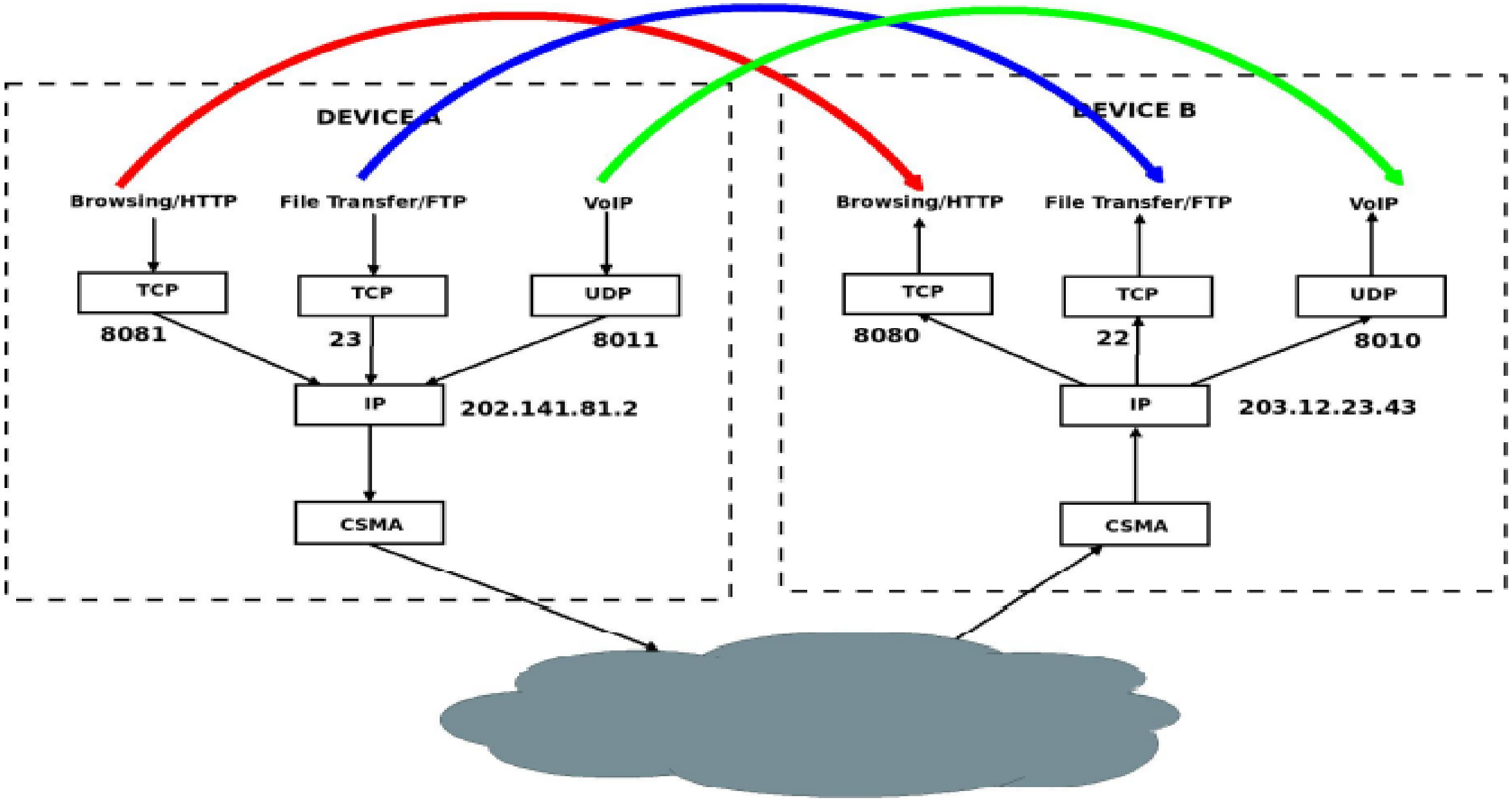
# Application Multiplexing in TCP/IP



# Application Multiplexing in TCP/IP

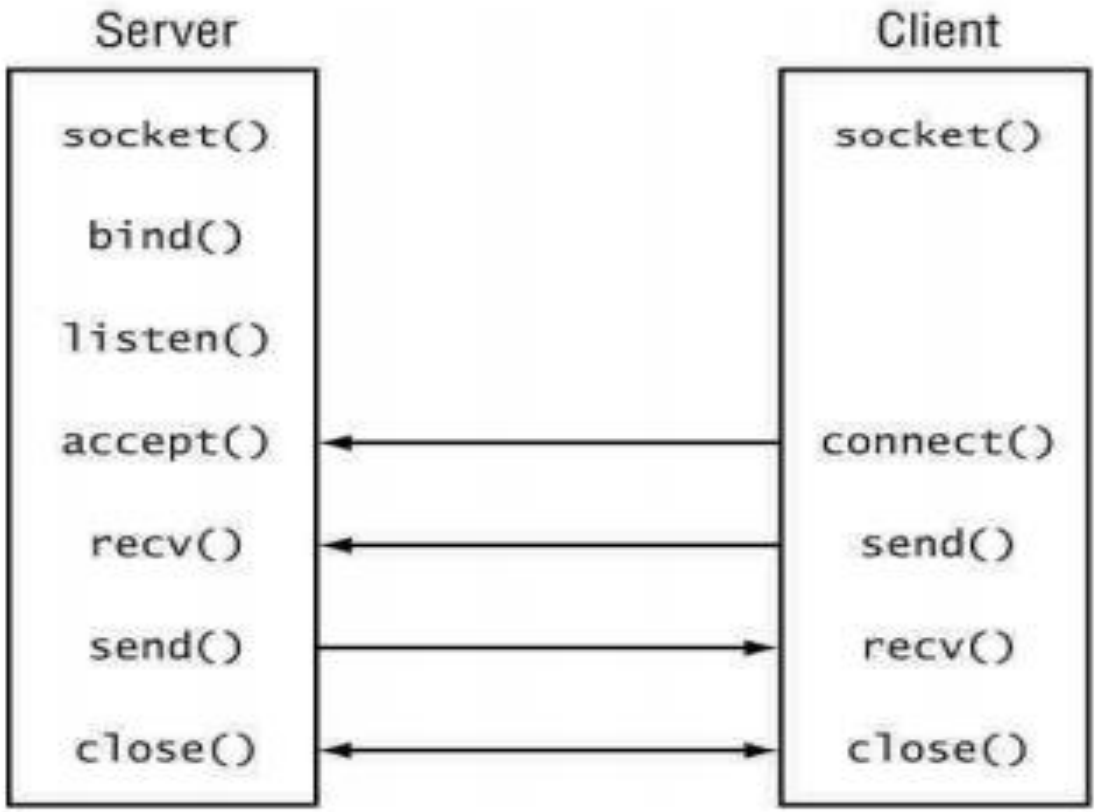


# What are Sockets?



# Socket Programming Framework/API

A set of **system calls** to get the service from TCP/IP protocol stack (`net` module in the OS kernel).



# Types of Sockets

- Transport layer supports two services - Reliable (TCP), and Unreliable (UDP)
- Two types of sockets:

1) Stream Socket (SOCK STREAM): Reliable, connection oriented (TCP based)
2) Datagram Socket (SOCK DGRAM): Unreliable, connection less (UDP based)

# Socket API

int s = socket(domain, type, protocol); - Create a socket

- domain: Communication domain, typically used AF INET (IPv4 Protocol)
- type: Type of the socket - SOCK STREAM or SOCK DGRAM
- protocol: Specifies protocols - usually set to 0.

- int status = bind(sockid, &addrport, size); - Reserves a port for the socket.
  - sockid: Socket identifier
  - addrport: struct sockaddr in - the (IP) address and port of the machine (address usually set to INADDR ANY chooses a local address)
  - size: Size of the sockaddr structure

# struct sockaddr _in

- sin_family : Address family, AF INET for IPv4 Protocol
- sin_addr.s addr: Source address, INADDR ANY to choose the local address
- sin_port: The port number
- We need to use htons() function to convert the port number from host byte order to network byte order.

```
struct sockaddr in serveraddr;
int port = 3028; serveraddr.sin family = AF INET;
serveraddr.sin addr.s addr = INADDR ANY;
serveraddr.sin port = htons(port);
```

# Listen and Accept a Socket Connection

```
struct sockaddr_in cli_addr;
listen(sockfd,5);
clilen = sizeof(cli addr);
newsockfd = accept(sockfd,(struct sockaddr *) &cli addr, &clilen);
```

Active Open and Passive Open

- The server needs to announce its address, remains in the open state and waits for any incoming connections - Passive Open
- The client only opens a connection when there is a need for data transfer - Active Open
- Connection is initiated by the client

# Data Transfer through Sockets

For SOCK_STREAM:

- read(newsockfd,buffer,255);
- write(newsockfd,“I got your message”,18);

For SOCK_DGRAM:

- recvfrom(sock,buf,1024,0,(struct sockaddr *)&from,&fromlen);
- sendto(sock,“Got your message”,17,0,(struct sockaddr *)&from,fromlen);