**High Level Design**

1. **Lecture 1**
2. **Network Protocols**
3. **What is Client Server Model**
4. **Peer to Peer Model**
5. **HTTP vs TCP vs UDP vs FTP vs SMTP Model**
6. **Examples**

**Network Protocols**

A network protocol is a set of rules that govern data communication between different devices in the network. It determines what is being communicated, how it is being communicated, and when it is being communicated. It permits connected devices to communicate with each other, irrespective of internal and structural differences.

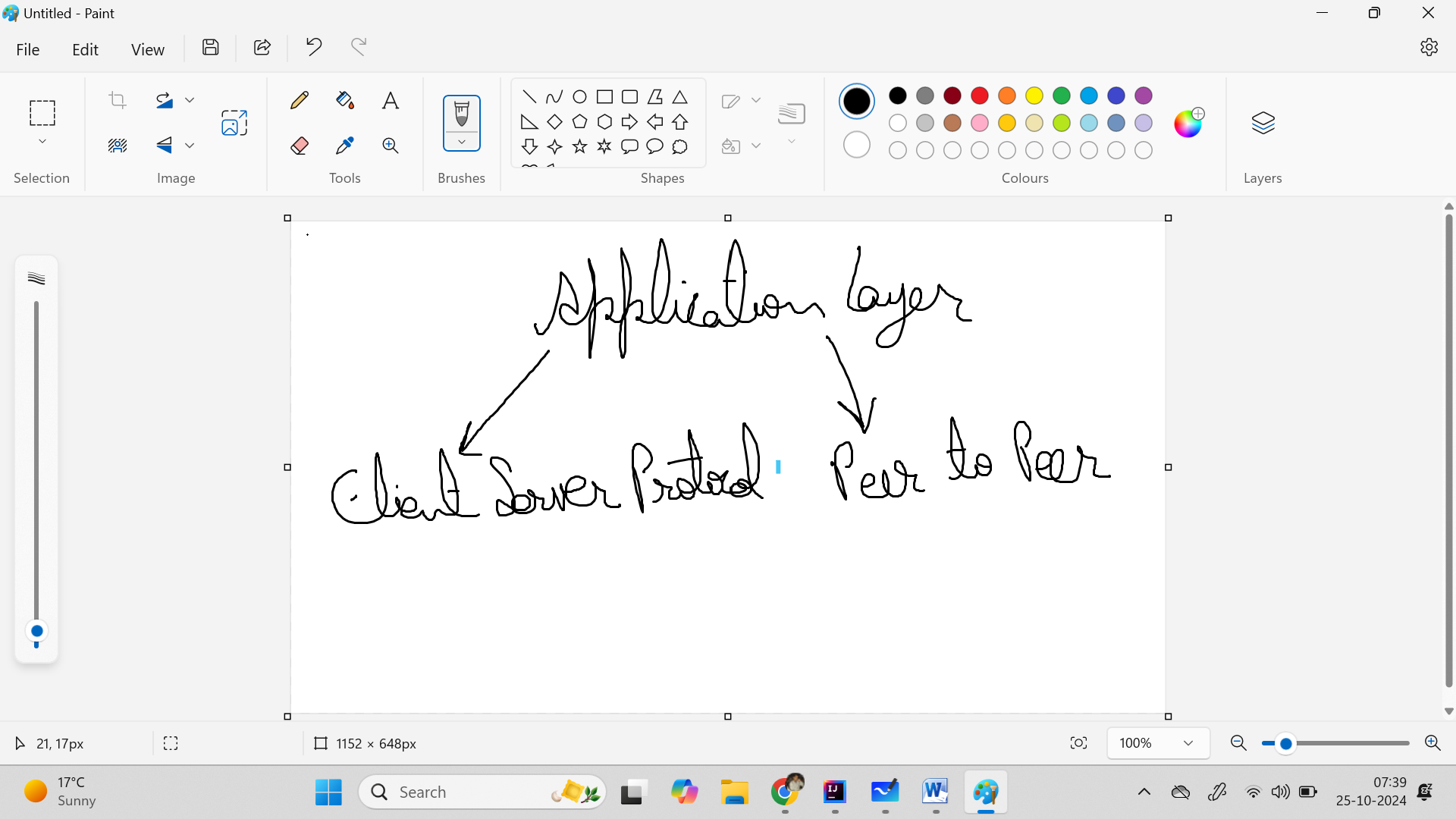
**There are 7 layers of protocols**

1. **Application Layer**
2. **Presentation Layer**
3. **Session Layer**
4. **Transport**
5. **Data Link Layer**
6. **Network Layer**

**We will be discussing 2 layers**

1. **Application Layer**
2. **Transport Layer**

**Application Layer**

****

**Client Server Protocols are**

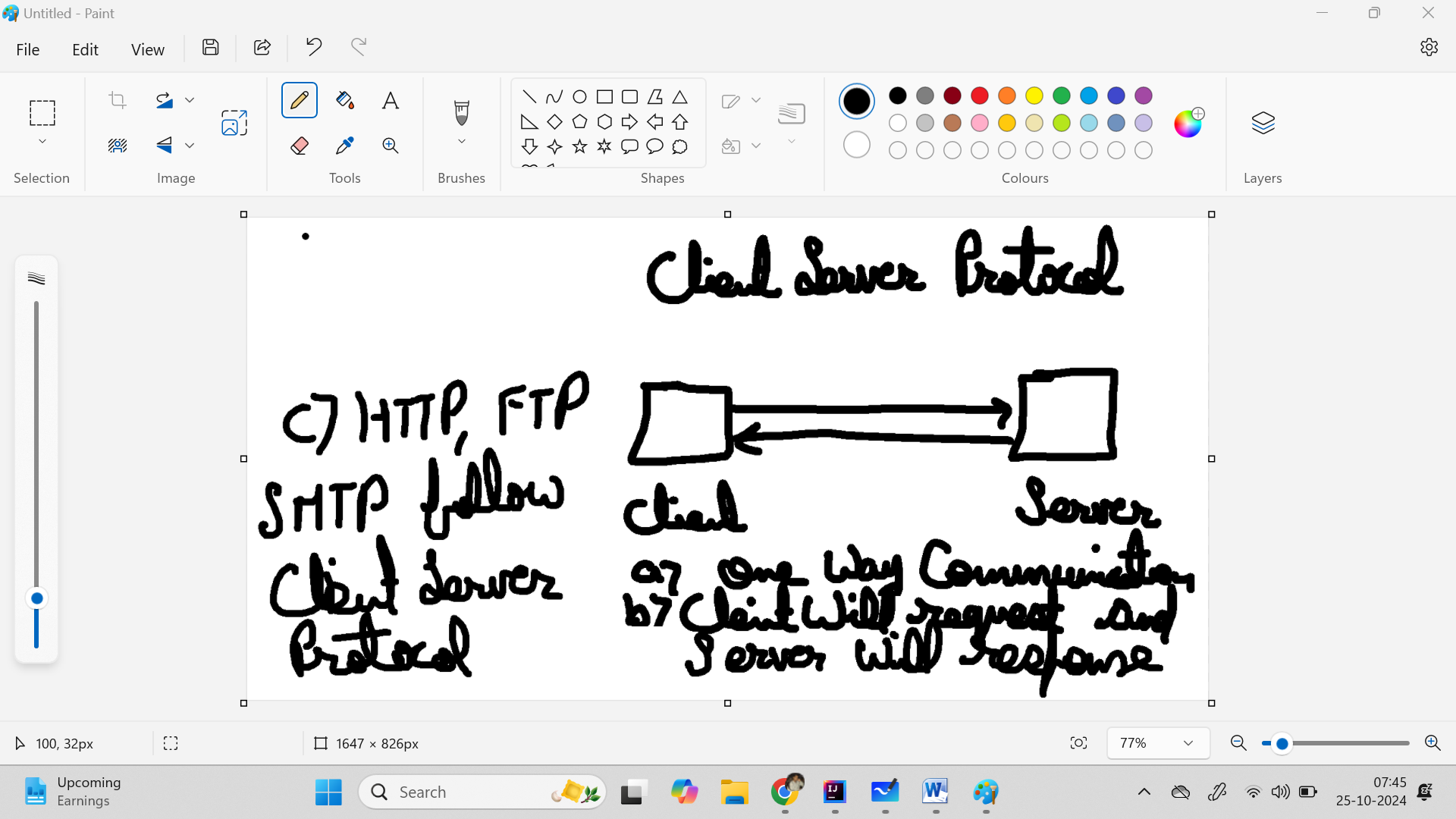
1. **HTTP**
2. **SMTP**
3. **FTP**
4. **Web Socket**

**Peer to Peer Protocol**

**Web RTC**

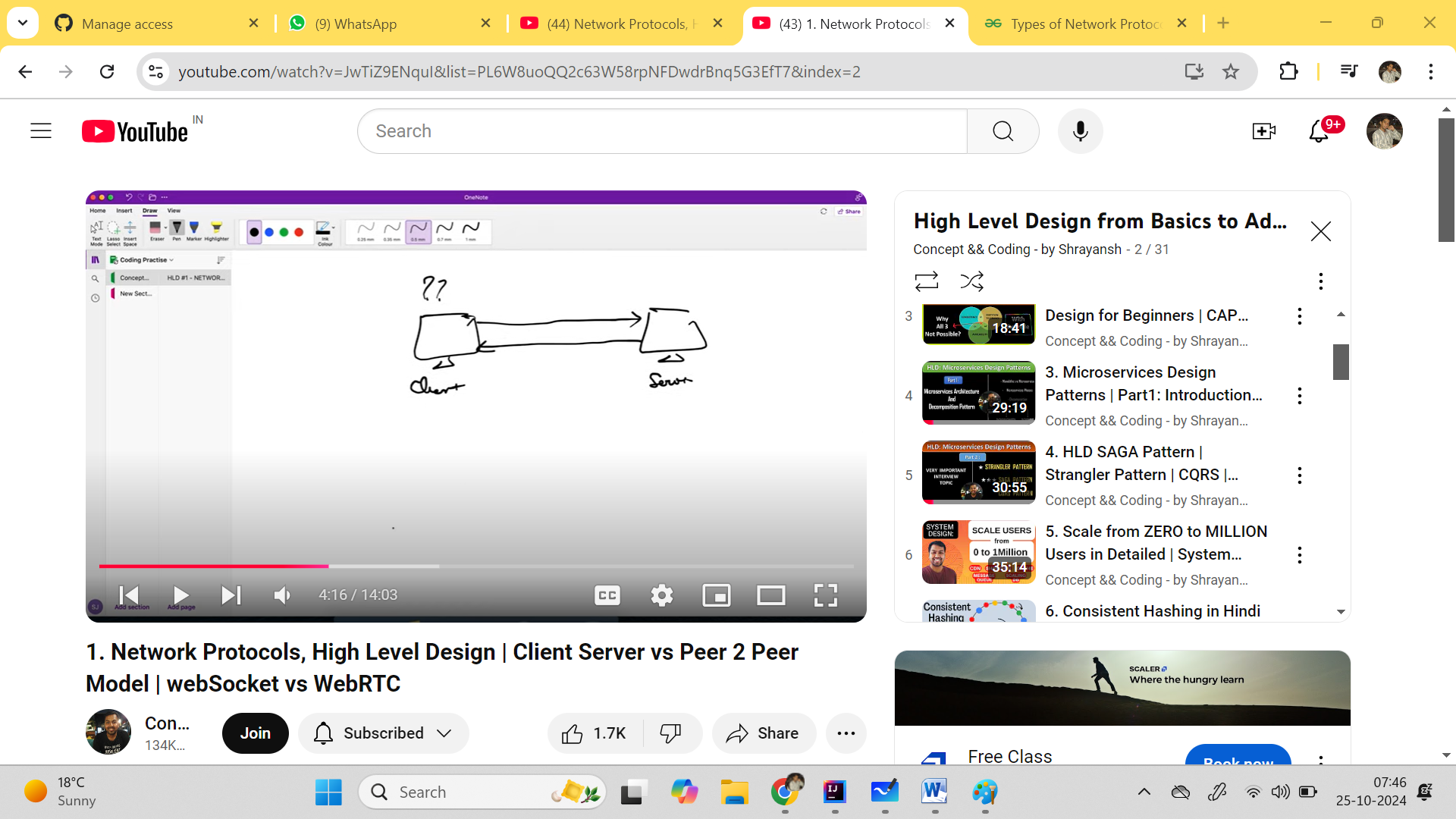
**Client Server Protocol**

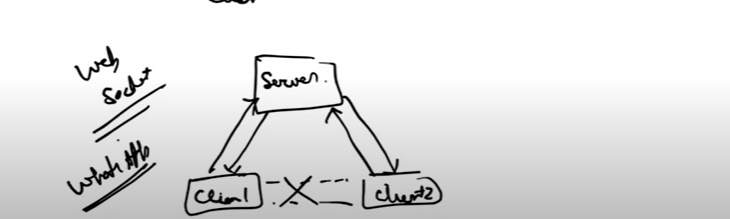
**HTTP, SMTP, FTP**

****

**Web Socket**

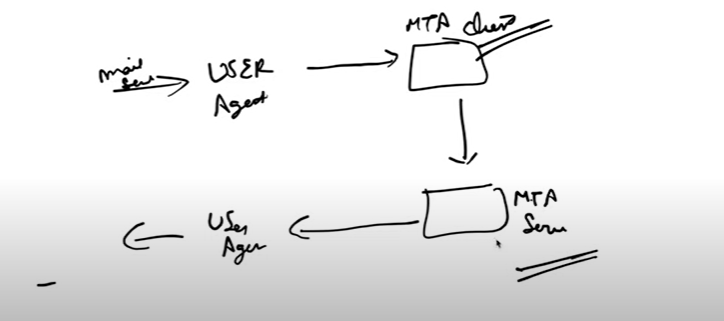
1. **Bidirectional communication**
2. **Client can talk to server and server can talk to client**

****

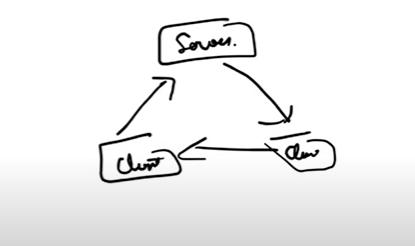
****

**We will use Web Socket when we have to develop the whtsapp, messaging app.**

1. **HTTP:-**
2. **create one connection**
3. **We have hypertext where we will jump to next page**
4. **FTP:-**
5. **Two connection created**
6. **A) Control Connection b) Data Connection**
7. **SMTP:-**
8. **Used for sending the mail and IMAP used to receiving the email**

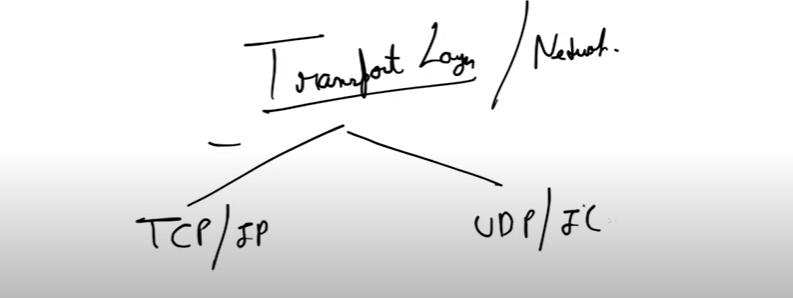
****

**Peer to Peer Protocol**

****

**WebRTC peer to peer protocol**

**Transport Layer Protocol**

****

1. **TCP/IP will create the connection**
2. **There will be ordering to send/recieive the message**
3. **TCP/IP will be using in messaging app**
4. **UDP/IP will create the multiple connection**
5. **No ordering will maintain and hence it is fast and will be using in the video calls**
6. **UDP/IP will be using in video calls**