Applications Layer (user interact GUI eg Firefox):

- The Application layer is the software tools that end user see and interact with (end user sees) eg: Firefox. Chrome. Should be transparent to most users
- Applications provides the means to generate and receive data to be transported over the network
- Two types of Application/Communication models
    - **Client and Server:**
        - Most/Traditional common communication model in modern data network
        - The use of dedicated servers provide content and services to end user devices (clients) eg: Murdoch lms on dedicated server provides the service to client
    - **Peer to Peer model:**
        - End user devices acts as both client and server to deliver data between the machines two end user devices (no servers)
        - They may still rely on some centralised services/servers eg: Bittorrent needs tracker

Application Layer Protocol Types (System of rules govern how messages flow across network):

- Domain Name System: Is a hierarchical system that translates the human readable domain name and associated ip address.
    - General top level domain: .com
    - CcTLD (country specific top level domain): .com.au
    - Uses UDP or TCP port 53
    - Why DNS uses UDP

DNS stands for domain name system. It is the process in which fetches the Ip address from the human readable domain.The DNS process uses packets of a small size (512kb). Therefore, if a fault occurs in the DNS process then we can just request another DNS process with is no big issue. Also in the application layer we can add extra security protect to help mitigate the decision of UDP over TCP. Also for the average person browsing the web the biggest desire is to load pages fast. So the use of UDP supports their desire.

- Hypertext Transfer Protocol (HTTP): retrieve (Transfer content from web to our pc) content from web pages/server Below possible responses when initiating a retrieval via Wireshark
    - HTTP Get: when you type in url then a request is sent
    - HTTP OK: Response given when the request succeeded
    - HTTP POST OR PUT: messages are used to send data to webservers

    - TCP port 80

- File Transfer Protocol:
    - Designed for transferring data rather than displaying. Downloading and uploading large file over internet
    - Uses two ports:
        - TCP port 20- for the transfer of data

        - TCP port 21- for command and control functions (establishing connection/navigation of file system)

- Dynamic Host Configuration:
    - Used to automatically get an Ip address from network. Can be manually assignged static (ip must remain same. When we want same IP) or dynamic
    - The devices such as home routers have DHCP active by default
    - Uses two ports:
        - UDP port 67- for any request sent by client
        - UDP port 68- for responses from server

- Telnet:
    - Provides an interactive command line that allows remote management of network devices and servers
    - Telnet is old protocol and transmit clear set
        - - Not secure and disabled in many operating systems
        - -Superseded by ssh
    - Uses TCP port 23

- Secure Shell (SSH):
    - Also provides interactive command line that allows remote management of devices and servers
    - However, it encrypts all sessions data to prevent third parties from reading any intercepted data
    - Uses TCP port 22

- Simple Mail Transfer Protocol:
    - Used for mainly for sending email between mail servers but can receive only between different mail servers
    - Users send email to client mail server (Gmail servers, outlook servers etc)
    - Uses port 25 or 465 587 (more secure)
    - Receiving Mail:
        - **Post Office Protocol (pop):** TCP 110. Bad since open email it no longer present
        - **Internet Message Access:** TCP 142

- Post office protocol: is an application layer internet standard protocol used by email clients to retrieve emails from servers. Emails come from servers not directly from people

***ADD? The process of Domain name Services?**

Presentation Layer (Operating system works on):

- Coding and conversion of application layer data to ensure that data from source can be interpreted by destination eg file formats: MPEG, JPG

Session Layer:

- The session layer handles the exchange of information to initiate, restart, keep alive and terminate conversations (between devices you are trying to get info from) eg: VPNS

Transport Layer:

- Deals how much data sent one time
- Segments application data into transportable chunks for transmission
- May reassemble segments and provides reliability
- Contain many different **transport layer protocols** to deal with different requirements for applications
- Uses **ports** to track individual conversations and identify applications (think usb port)

Ports (Think like usb port):

- They are a 16 digit Integer value (0-65535) in which the operating system uses for identification of applications. Some application use a specific port they must use and some have multiple ports they can connect to ie: (My website uses port 3000 so once it's been used new port is needed. Again think usb port)
- Ports can be classified:
    - **Well known port (0 -1023):** Common services and applications
        - **HTTP:** 80
        - **SMTP:** 20
    - **Registered ports (1024-49151):** commonly used services and applications
        - Open vpn- 1194
    - **Dynamic/Private port (49152-65535):** for client initiated sessions

Transport Layer Protocols: (Uses packets)

Transmission Control Protocol (TCP): (reliable) File download + Loading websites

- TCP is a **connection-oriented protocol and reliable**

- Connection oriented means- Communication between two devices must be clearly *started* and *terminated* for reliability and connection via three way handshake. EG- **Email**
- Used in safety and critical applications / services such as website that need to run all the time
- It guarantees packet delivery such for critical application such as email. But will have additional delay.
- IF TCP was implemented before then we can't change it

***Imagine switched network WHERE YOU HAVE TWO HOST SENDING PACKETS USING TRANSVERESE PATHS.

Properties that make TCP Reliable:

Context: When transmitting packers over network some packets make takes different paths and therefore segments may come to destination in different order (due to faults like power cut) which is bad

- Offers ***In-Order delivery*** of segments to destination, through the process of **sequence numbers**. If a fault causes segments to come in different orders then TCP allows it to be reordered correctly.
- All data transmitted using TCP must be acknowledge (so it keeps track of packets sent)
- When segments are not acknowledged they are retransmitted by the sender so every segment is eventually received
- Has **congestion control and flow control**
    - This is a mechanism, which manages rate at which tcp connection sends packets. Without congestion control the tcp connection would overflood router with packets leading to **buffering,** which leads to **congestion** and ***eventually the router had to drop packets causing→ packet loss.***
    - Specifies the congestion windows, which determines the number of unacknowledged segments that can be in flight to the destination at a window.
    - Has two ways to implement congestion control with newReno

Bufferbloat:

- The result of implementing idea that we can increase the buffer size so it can take more packets.
- What occurs is that newreno will always fill buffers up to capacity and therefore the issue of packets being sent slower then received is still an issue!

ACK are cumulative so it acknowledges preceding segments

Receiver acknowledges next expected byte

TCP sequence numbers are used in conjunction with ack

*Explain New reno + flow control?

- This processes of making it clear start there is communication is **three-way handshake ie when you enter website it initiates a connection between device and server (3 packets send back and forth):**
    - Client transmit SYN **(synchronisation request)** to host
    - Receiver transmit Syn-ack **(Synchronisation acknowledgement)** to respond to acknowledge
    - Client acknowledges again to establish connection ack **(acknowledgement)**
- The Process of clearly terminated connection is: Four step process (7 Packets)
    - Client sends FIN (finishing packet)
    - Receiver sends ACK
    - Receiver send FIN
    - Client sends ACK

User Datagram Protocol:

- UDP is **connectionless** and therefore **unreliable**.
    - The protocol used to deliver data **timely** send packets when available **but unreliable**

- It is connectionless meaning that there is no verifying if gets to destination
- Not necessary less reliable but no guarantee or verifying if gets to destination
- No congestion or lower overhead (less steps) why? Because it doesn't initiate a teardown process such as a three way handshake
    - Eg Voice and video communication since if we retransmit packet then voice conversation gibberish or game turns up seconds later

- **Why/Purpose** use UDP
    - Because resent data is useless and additional delays should be avoided
        - Skype/Telecommunication: Additional delays for retransmission is not ideal as you want real time conversations
        - Gaming- again is not ideal to resend packets after action already happened
    - TCP is very complex or has too many overheads
        - Full TCP implementation may be too complex for some machines with low ram, or slow cpu
        - Alternatives can use UDP + simpler acknowledgement scheme which is more efficient potentially
    - UDP is connectionless and will not setup and teardown connection like TCP
        - Setting up and tearing down a TCP connection requires a min of 6 packets uses lots of bandwidth
        - Uses lots of ram and cpu

Examples of where application may use UDP is media streaming since a loss of frame packets are no big deal

Examples of where application may use TCP is Email since we often require all data and packets to be sent to the destination. And since email services are not too affected by the fact TCP takes longer to send data because Emails were not designed for immediate communication

ADD QUIZZ

Describe the method used by operating systems to differentiate between TCP connections. Use an example to illustrate how an operating system would determine that a TCP synchronisation acknowledgement (SYN-ACK) is in response to a particular synchronisation request (SYN).