- The OSI (Open Systems Interconnection) model is a conceptual model developed by the International Organization for Standardization (ISO) that describes how communications should occur in a computer network. In other words, the OSI model defines a framework for computer network communications. Although this model is theoretical, it is vital to learn and understand as it helps grasp networking concepts on a deeper level. The OSI model is composed of seven layers

  The numbering starts with the physical layer being layer 1, while the top layer, the application layer, is layer 7. To help you remember the layers from bottom to top, you can use a mnemonic such as "Please Do Not Throw Spinach Pizza Away." or "All People Seem To Need Data Processing ".

  Remembering the OSI model layers with their layer numbers is important; otherwise, you will struggle to understand terms such as "layer 3 switch" or "layer 7 firewall."

  The **OSI** model (or **O**pen **S**ystems **I**nterconnection Model) refers to the **Open System Interconnection** reference model.

- We use this OSI model to describe the process that data takes it traverses our networks.

- The OSI (Open System Interaction) model is not designed to be detailed description of model but instead intended to describe a broad overview of data traverses our systems.

- We're also not describing the OSI protocol suit! Most of the Protocols we use today based on TCP/IP. The OSI model is one that we can apply to many different protocols and work perfectly with the TCP/IP model that we use today.

- There are many different protocols that might operate at an induvial layer of the OSI model. A layer can have many protocol that might exist and work at particular layer.

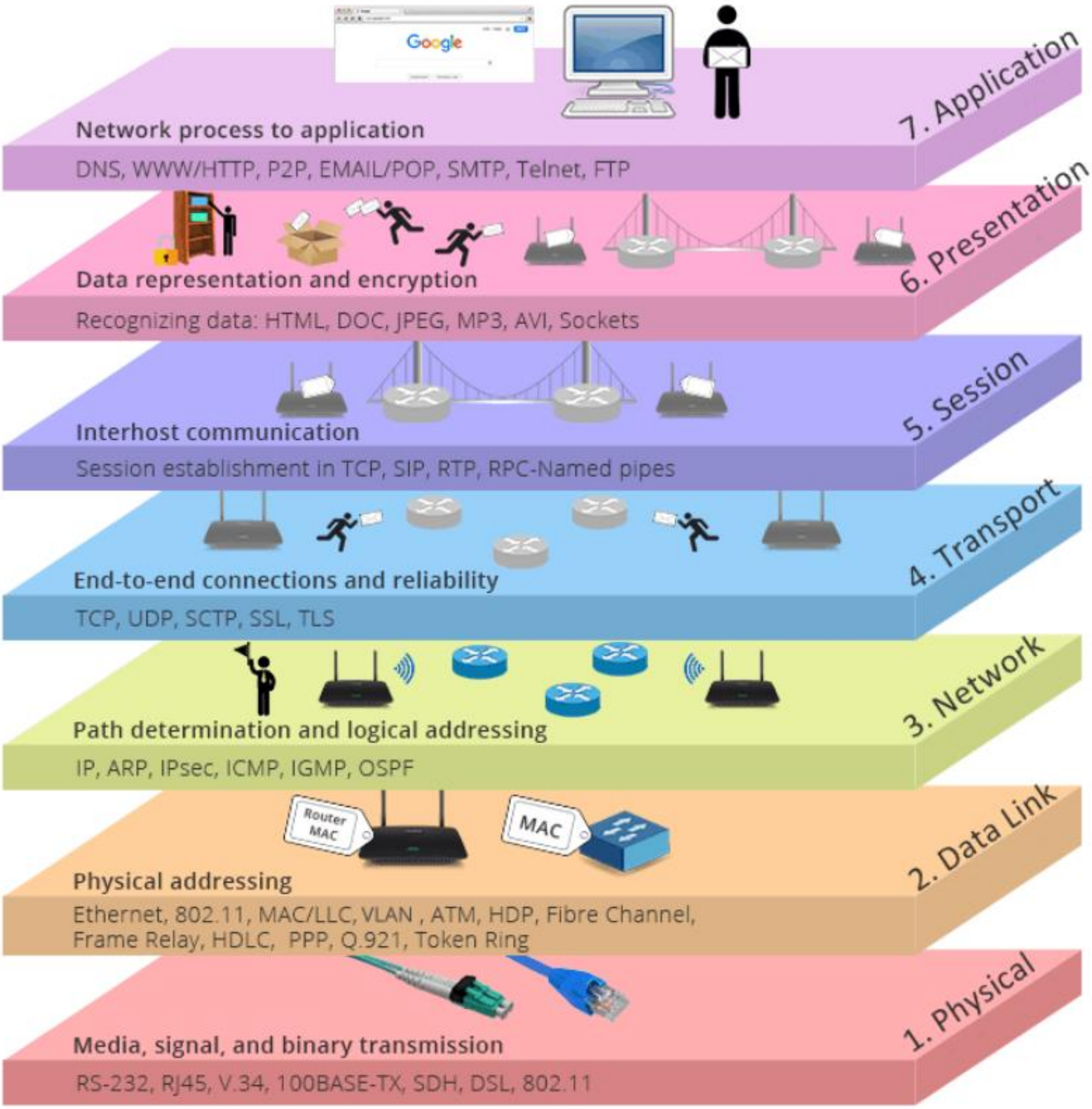
- Understanding OSI model we can converse with other people in information technologies in a way that we would all understand!

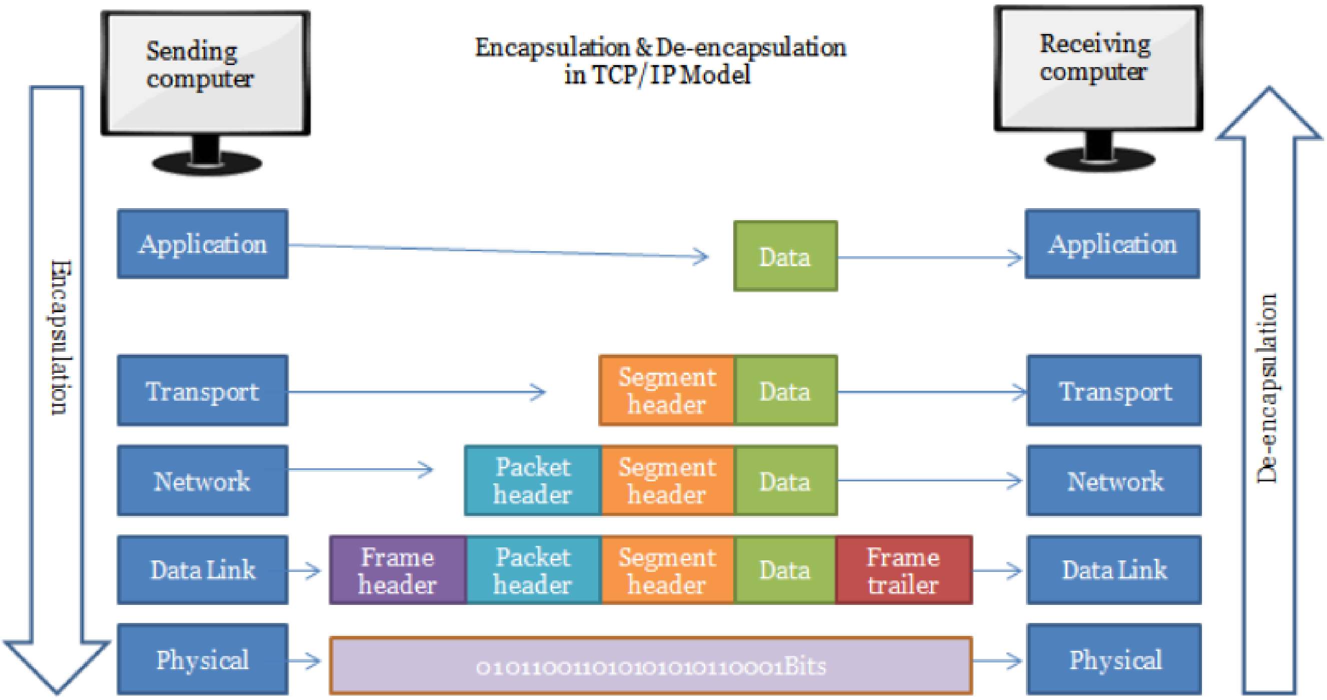
- OSI model provides a framework dictating how all networked devices will send, receive and interpret data.

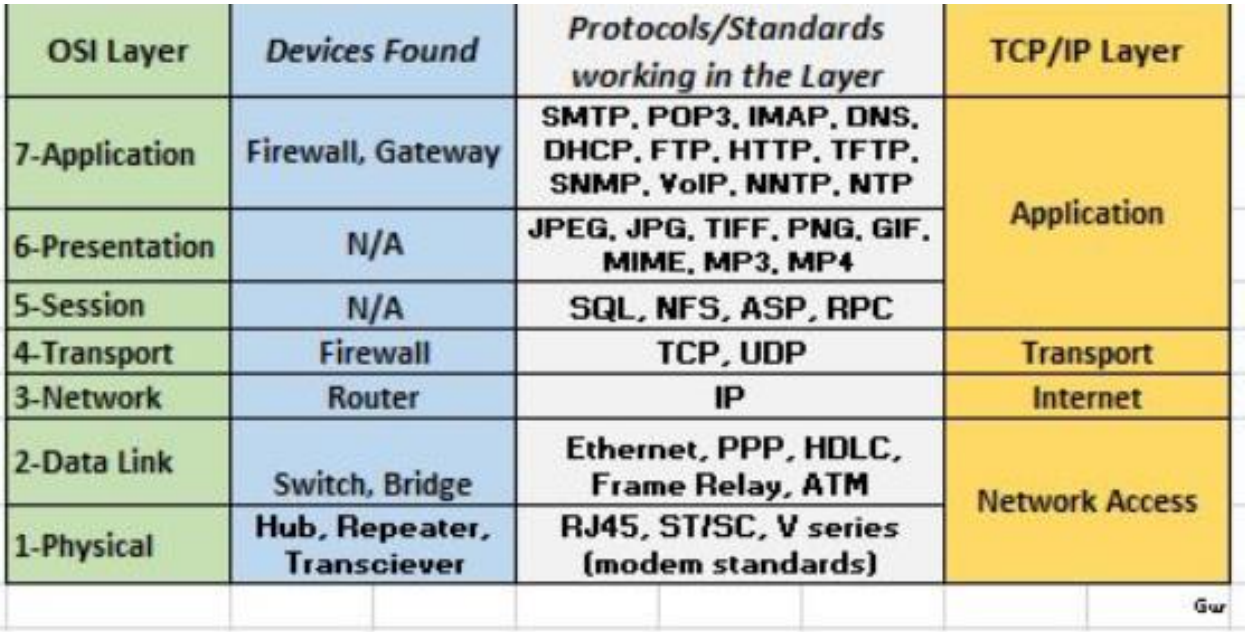
- One of the main benefits of the OSI model is that devices can have different functions and designs on a network while communicating with other devices. Data sent across a network that follows the uniformity of the OSI model can be understood by other devices.

- At every individual layer that data travels through, specific processes take place, and pieces of information are added to this data that this process is called encapsulation.

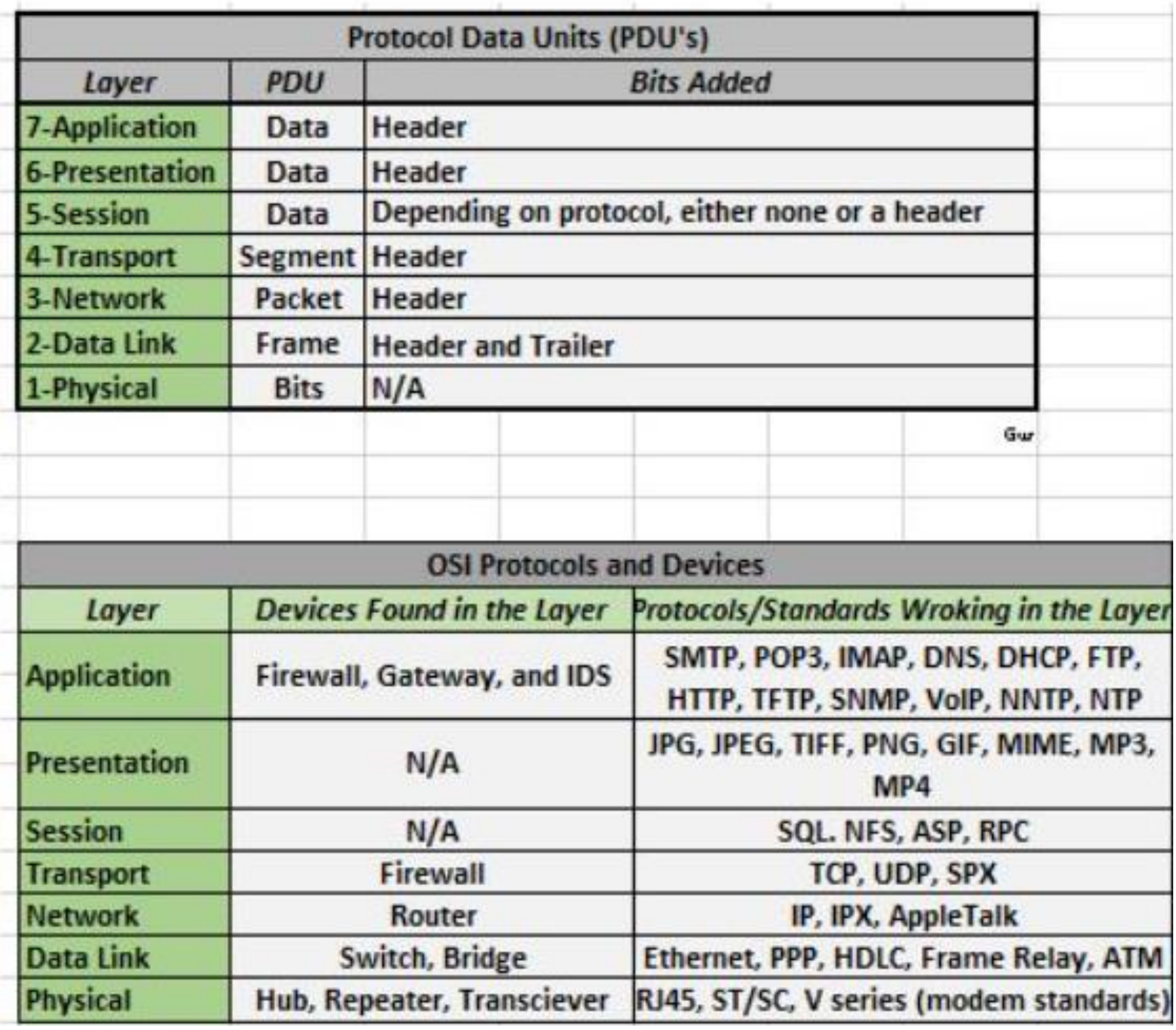
- TCP/IP model is standardize computer networking, it is same description as the OSI model but OSI model world widely isn't used on the real work the TCP/IP model is however real deal!


Google
7. Application
Network process to application
DNS, WWW/HTTP, P2P, EMAIL/POP, SMTP, Telnet, FTP
6. Presentation
Data representation and encryption
Recognizing data: HTML, DOC, JPEG, MP3, AVI, Sockets
5. Session
Interhost communication
Session establishment in TCP, SIP, RTP, RPC-Named pipes
4. Transport
End-to-end connections and reliability
TCP, UDP, SCTP, SSL, TLS
3. Network
Path determination and logical addressing
IP, ARP, IPsec, ICMP, IGMP, OSPF
2. Data Link
Router MAC
MAC
Physical addressing
Ethernet, 802.11, MAC/LLC, VLAN , ATM, HDP, Fibre Channel,
Frame Relay, HDLC, PPP, Q.921, Token Ring
1. Physical
Media, signal, and binary transmission
RS-232, RJ45, V.34, 100BASE-TX, SDH, DSL, 802.11


Sending computer
Encapsulation & De-encapsulation in TCP/IP Model
Receiving computer
Encapsulation
Application
Data
Application
Transport
Segment header
Data
Transport
Network
Packet header
Segment header
Data
Network
Data Link
Frame header
Packet header
Segment header
Data
Frame trailer
Data Link
Physical
010110011010101010110001Bits
Physical
De-encapsulation



OSI Layer	*Devices Found*	*Protocols/Standards working in the Layer*	TCP/IP Layer
7-Application	Firewall, Gateway	SMTP, POP3, IMAP, DNS, DHCP, FTP, HTTP, TFTP, SNMP, VoIP, NNTP, NTP	Application
6-Presentation	N/A	JPEG, JPG, TIFF, PNG, GIF, MIME, MP3, MP4	
5-Session	N/A	SQL, NFS, ASP, RPC	
4-Transport	Firewall	TCP, UDP	Transport
3-Network	Router	IP	Internet
2-Data Link	Switch, Bridge	Ethernet, PPP, HDLC, Frame Relay, ATM	Network Access
1-Physical	Hub, Repeater, Transciever	RJ45, ST/SC, V series (modem standards)	



Protocol Data Units (PDU's)		
Layer	*PDU*	*Bits Added*
7-Application	Data	Header
6-Presentation	Data	Header
5-Session	Data	Depending on protocol, either none or a header
4-Transport	Segment	Header
3-Network	Packet	Header
2-Data Link	Frame	Header and Trailer
1-Physical	Bits	N/A

OSI Protocols and Devices		
Layer	*Devices Found in the Layer*	*Protocols/Standards Wroking in the Layer*
Application	Firewall, Gateway, and IDS	SMTP, POP3, IMAP, DNS, DHCP, FTP, HTTP, TFTP, SNMP, VoIP, NNTP, NTP
Presentation	N/A	JPG, JPEG, TIFF, PNG, GIF, MIME, MP3, MP4
Session	N/A	SQL. NFS, ASP, RPC
Transport	Firewall	TCP, UDP, SPX
Network	Router	IP, IPX, AppleTalk
Data Link	Switch, Bridge	Ethernet, PPP, HDLC, Frame Relay, ATM
Physical	Hub, Repeater, Transciever	RJ45, ST/SC, V series (modem standards)

Physical Layer

This layer is one of the easiest layers to grasp. Put simply, this layer references the physical components of the hardware used in networking and is the lowest layer that you will find. Devices use electrical signals to transfer data between each other in a binary numbering system (1's and 0's).

- Signalling, cabling, HUB, Modem NIC Wireless access Point.
- Physical layer don't have protocols to speak up because Physical layer makes sure ones and zeros get from different hosts
- When we refer to physical layer problem with network we are referring to Open System Interaction model's layer 1.
    - Fixing cabling, punch downs etc
    - Making trouble shooting such as run loopbacks tests, test/replace cables, swap adapter cards.

Data Link Layer

The data link layer focuses on the physical addressing of the transmission. It receives a packet from the network layer (including the IP address for the remote computer) and adds in the physical **MAC** (Media Access Control) address of the receiving endpoint (be aware encapsulation). Inside every network-enabled computer is a **N**etwork **I**nterface Card (**NIC**) which comes with a unique MAC address to identify it.

MAC addresses are set by the manufacturer and literally burnt into the card; they can't be changed – although they can be spoofed. When information is sent across a network, it's actually the physical address that is used to identify where exactly to send the information.

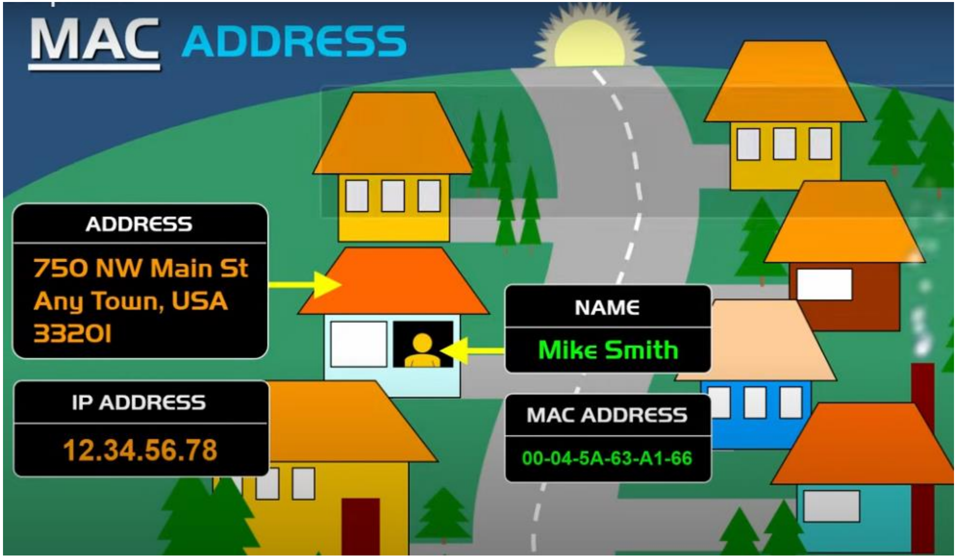
Additionally, it's also the job of the data link layer to present the data in a format suitable for transmission.

- Data Link Layer job is the allow individual systems to be addressed in such a way that ethernet frames get to right spot.

- Data Link Layer used to communicate between two devices on the network, that often refer to this as Media Access Control (MAC) because that is the Data Link Control Layer commonly associated wit the network cards that in our devices. Most of the time they are ethernet adapters or wireless adapters. We refer to that physical address on that device as the data link control address pr the MAC (Media Access Control).

- Since the network switches that we use on our network determine how to forward traffic based on destination MAC address. This layer we often refer to as the switching layer.

- Switches, Bridge, NIC, Wireless access points
- Mac address is used to identify device and IP address is used to locate device be aware of differences.



ARP

Devices can have two identifiers: A MAC address and an IP address, the **A**ddress **R**esolution **P**rotocol or **ARP** for short, is the technology that is responsible for allowing devices to identify themselves on a network.

Simply, the ARP protocol allows a device to associate its MAC address with an IP address on the network. Each device on a network will keep a log of the MAC addresses associated with other devices.

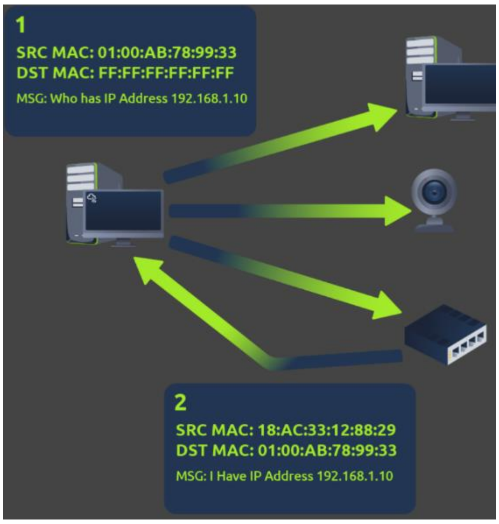
When devices wish to communicate with another, they will send a broadcast to the entire network searching for the specific device. Devices can use the ARP protocol to find the MAC address (and therefore the physical identifier) of a device for communication.

How does ARP Work?

Each device within a network has a ledger to store information on, which is called a cache. In the context of the **ARP** protocol, this cache stores the identifiers of other devices on the network.

In order to map these two identifiers together (IP address and MAC address), the ARP protocol sends two types of messages:

1. **ARP Request**
2. **ARP Reply**

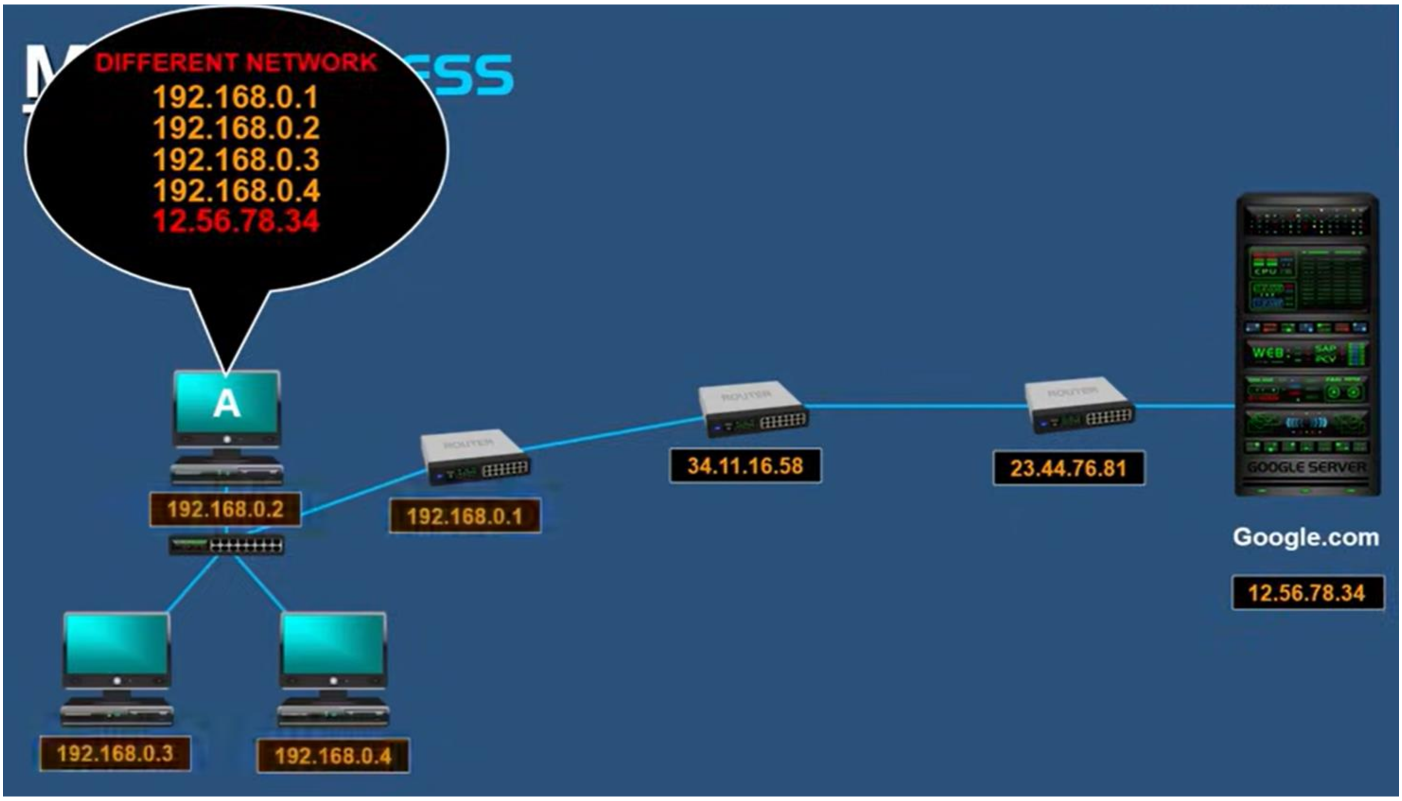

1
SRC MAC: 01:00:AB:78:99:33
DST MAC: FF:FF:FF:FF:FF:FF
MSG: Who has IP Address 192.168.1.10
2
SRC MAC: 18:AC:33:12:88:29
DST MAC: 01:00:AB:78:99:33
MSG: I Have IP Address 192.168.1.10

When an **ARP request** is sent, a message is broadcasted on the network to other devices asking, "What is the mac address that owns this IP address?" When the other devices receive that message, they will only respond if they own that IP address and will send an **ARP reply** with its MAC address. The requesting device can now remember this mapping and store it in its **ARP cache** for future use.

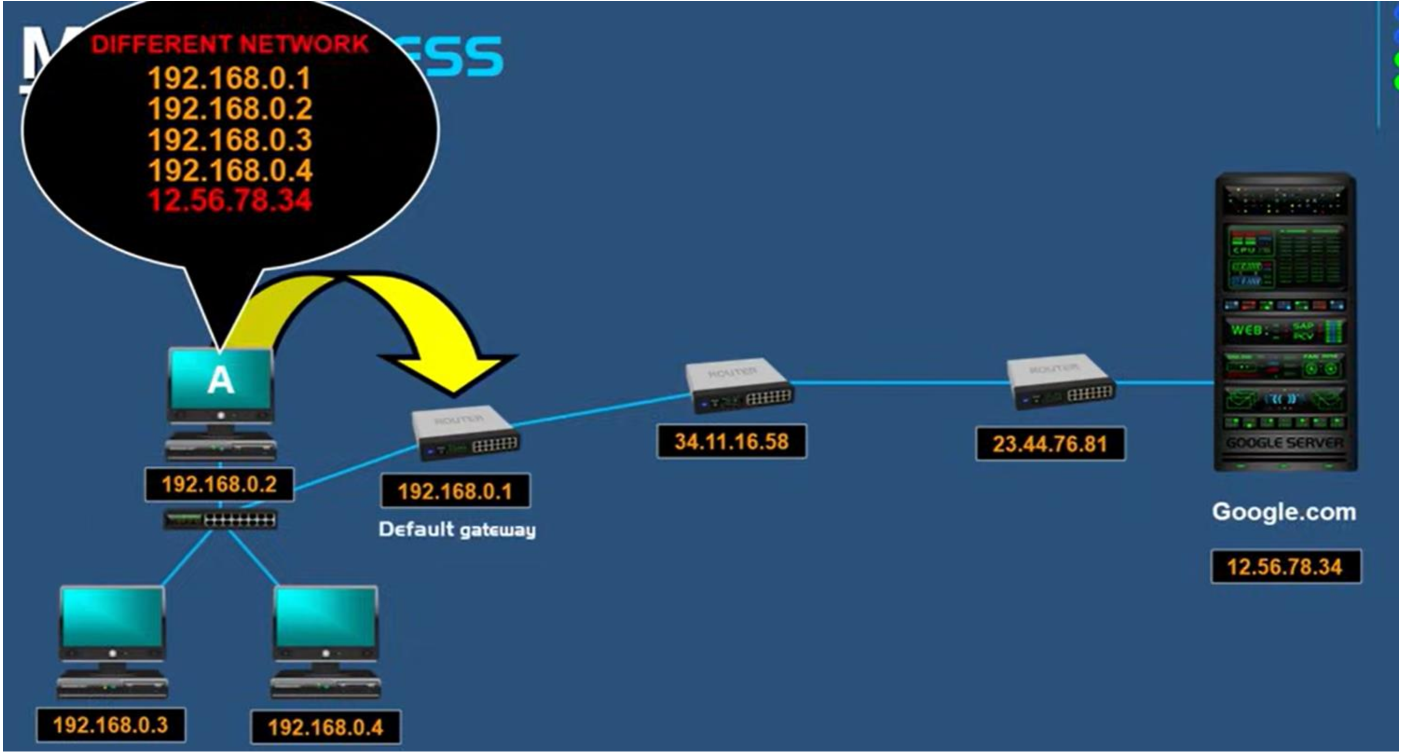
"ARP sends an Ethernet frame called an ARP request to every host on the network. This is called a broadcast ... The ARP request contains the IP address on the destination host ... and is the request 'if you are the owner of this IP address, please respond to me with your hardware address'. The destination host's ARP layer receives this broadcast, recognizes that the sender is asking for its hardware address, and replies with an ARP reply. This reply contains the IP address and the corresponding hardware address. The ARP reply is received and the IP datagram that forced the ARP request-reply to be exchanged can now be sent" (Stevens, TCP/IP Illustrated vol I, 54-55). The L3 lookup is only a normal operation of IP protocol stack (de-encapsulation).

Think like you will write google.com on web browser to order to go the google.com. It needs the MAC address for google's web server but problem is, google server doesn't know what it is. So this case your computer needs IP address to get to webserver MAC address.

So when you write google.com DNS will change into ip address 12.56.78.34 so your computer inspect google address and realize that IP is not in this location because the IP address is not in the same group as its local network.



So now computer A realizes that Google's Ip address is on different network and since it's on another network it's going to forward the data to default gateway which is the router MAC Address Explained



Network Layer

The third layer of the OSI model (network layer) is where the magic of routing & re-assembly of data takes place (from these small chunks to the larger chunk). Firstly, routing simply determines the most optimal path in which these chunks of data should be sent.

Whilst some protocols at this layer determine exactly what is the "optimal" path that data should take to reach a device, we should only know about their existence at this stage of the networking module. Briefly, these protocols include **OSPF** (**O**pen **S**hortest **P**ath **F**irst) and **RIP** (**R**outing **I**nformation **P**rotocol). The factors that decide what route is taken is decided by the following:

- What path is the shortest? I.e. has the least amount of devices that the packet needs to travel across.
- What path is the most reliable? I.e. have packets been lost on that path before?
- Which path has the faster physical connection? I.e. is one path using a copper connection (slower) or a fibre (considerably faster)?

At this layer, everything is dealt with via IP addresses such as 192.168.1.100. Devices such as routers capable of delivering packets using IP addresses are known as Layer 3 devices — because they are capable of working at the third layer of the OSI model.

- Network layer often refers as Routing Layer because this is the layer that router use to determine how to forward traffic.
- Ip destination address determines what is the next hop might be for traffic traversing the network.
- The Network layer also fragment these frames into multiple pieces especially if we're sending it across a network that may require smaller frames than what is on our local network. So we can cut those frames into smaller pieces to be able to fit them through the network and put those pieces back together on the other side (destination).
- Anytime we are referring to Ip addressing (Internet protocol), subnet masks anything related to an IP addressing, or anything about routing then probably referring to layer 3 (network layer).

Transport Layer

Layer 4 of the OSI model plays a vital part in transmitting data across a network and can be a little bit difficult to grasp. When data is sent between devices, it follows one of two different protocols that are decided based upon several factors:

- TCP
- UDP

TCP

Transmission **C**ontrol **P**rotocol (**TCP**). Potentially hinted by the name, this protocol is designed with reliability and guarantee in mind. This protocol reserves a constant connection between the two devices for the amount of time it takes for the data to be sent and received.

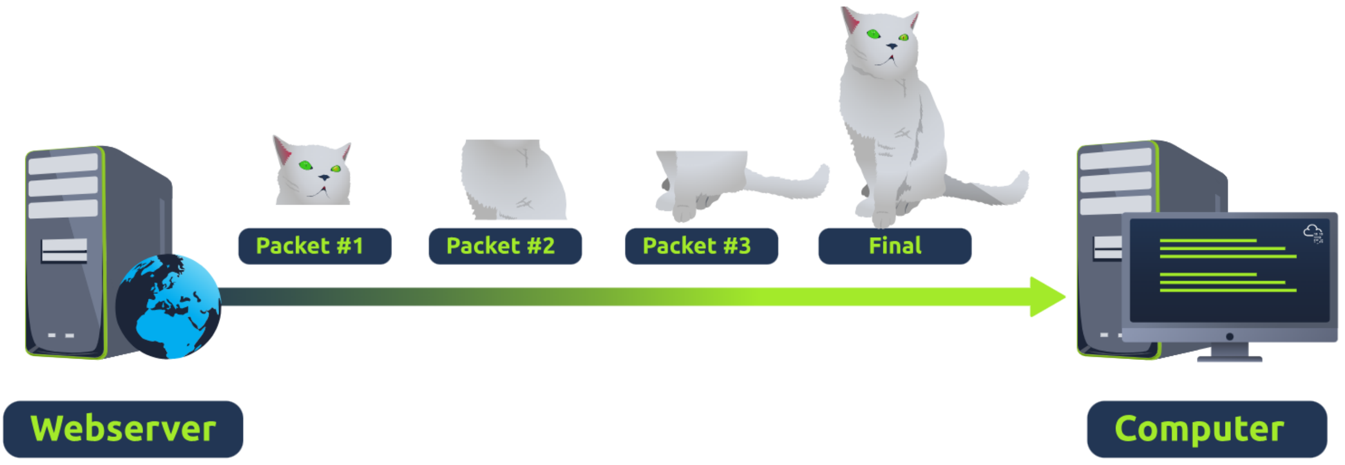
Not only this, but TCP incorporates error checking into its design. Error checking is how TCP can guarantee that data sent from the small chunks in the session layer (layer 5) has then been received and reassembled in the same order.

Let's summarise the advantages and disadvantages of TCP in the table below:

Advantages of TCP	**Disadvantages of TCP**
Guarantees the accuracy of data.	Requires a reliable connection between the two devices. If one small chunk of data is not received, then the entire chunk of data cannot be used.
Capable of synchronizing two devices to prevent each other from being flooded with data.	A slow connection can bottleneck another device as the connection will be reserved on the receiving computer the whole time.
Performs a lot more processes for reliability.	TCP is significantly slower than UDP because more work has to be done by the devices using this protocol.

TCP is used for situations such as file sharing, internet browsing or sending an email. This usage is because these services require the data to be accurate and complete (no good having half a file!).

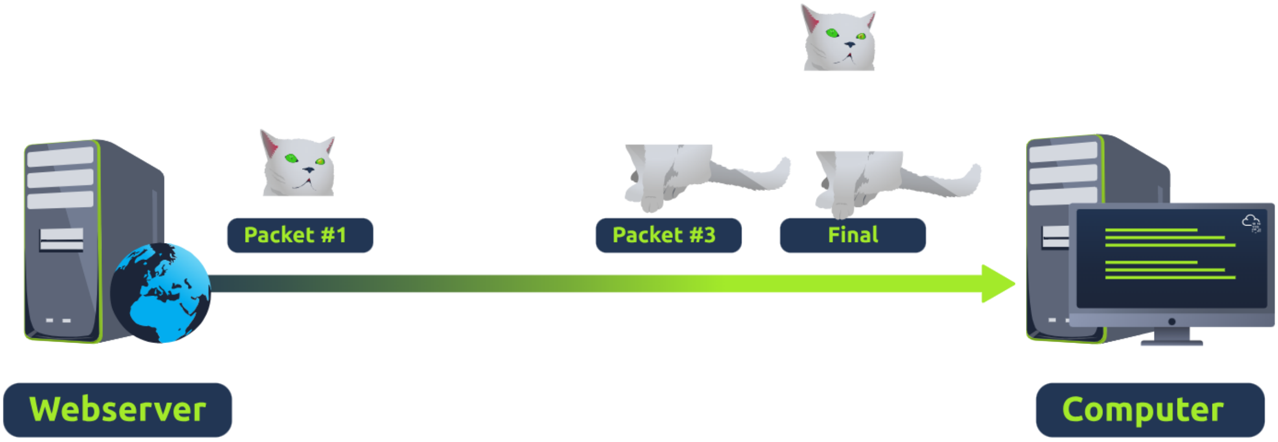
In the diagram below, we can see how a picture of a cat is broken down into small pieces of data (known as packets) from the "webserver", where the "computer" re-constructs the picture of the cat into the correct order.



UDP (**U**ser **D**atagram **P**rotocol)

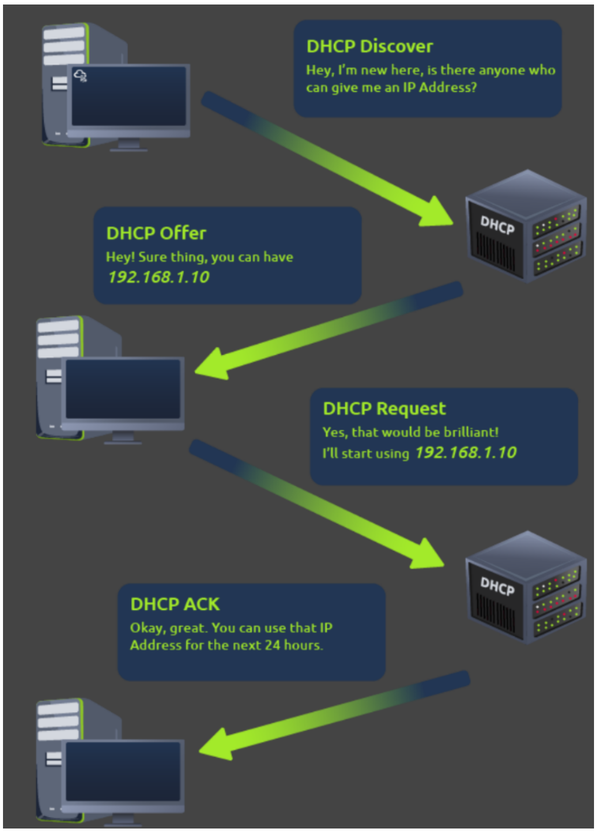
This protocol is not nearly as advanced as its brother - the TCP protocol. It doesn't boast the many features offered by TCP, such as error checking and reliability. In fact, any data that gets sent via UDP is sent to the computer whether it gets there or not. There is no synchronisation between the two devices or guarantee; just hope for the best, and fingers crossed.

Advantages of UDP	**Disadvantages of UDP**
UDP is much faster than TCP.	UDP doesn't care if the data is received.
UDP leaves the application layer (user software) to decide if there is any control over how quickly packets are sent.	It is quite flexible to software developers in this sense.
UDP does not reserve a continuous connection on a device as TCP does.	This means that unstable connections result in a terrible experience for the user.



UDP is useful in situations where there are small pieces of data being sent. For example, protocols used for discovering devices (***ARP*** and ***DHCP)*** or **larger files such as video streaming** (where it is okay if some part of the video is pixelated. Pixels are just lost pieces of data!

DHCP


DHCP Discover
Hey, I'm new here, is there anyone who can give me an IP Address?
DHCP
DHCP Offer
Hey! Sure thing, you can have 192.168.1.10
DHCP Request
Yes, that would be brilliant! I'll start using 192.168.1.10
DHCP
DHCP ACK
Okay, great. You can use that IP Address for the next 24 hours.

IP addresses can be assigned either manually, by entering them physically into a device, or automatically and most commonly by using a **DHCP** (**D**ynamic **H**ost **C**onfiguration **P**rotocol) server. When a device connects to a network, if it has not already been manually assigned an IP address, it sends out a request (DHCP Discover) to see if any DHCP servers are on the network. The DHCP server then replies back with an IP address the device could use (DHCP Offer). The device then sends a reply confirming it wants the offered IP Address (DHCP Request), and then lastly, the DHCP server sends a reply acknowledging this has been completed, and the device can start using the IP Address (DHCP ACK).

Session

Before we can send that information from one side of the network to the other we may need to create a session so that a device is able to receive a data. Session layer provides communication management between point A to point B. Anything relating to the initiation of a session, stopping the session or restarting a session.

If an application is using some type of control protocol or your tunnelling information within existing data then you are probably using OSI layer 5.

The session layer is also responsible for closing the connection if it hasn't been used in a while or if it is lost. Additionally, a session *can* contain "checkpoints," where if the data is lost, only the newest pieces of data are required to be sent, saving bandwidth.

What is worthy of noting is that sessions are unique — meaning that data cannot travel over different sessions, but in fact, only across each session instead.

Presentation

Layer 6 of the OSI model is the layer in which standardisation starts to take place. Because software developers can develop any software such as an email client differently, the data still needs to be handled in the same way — no matter how the software works.

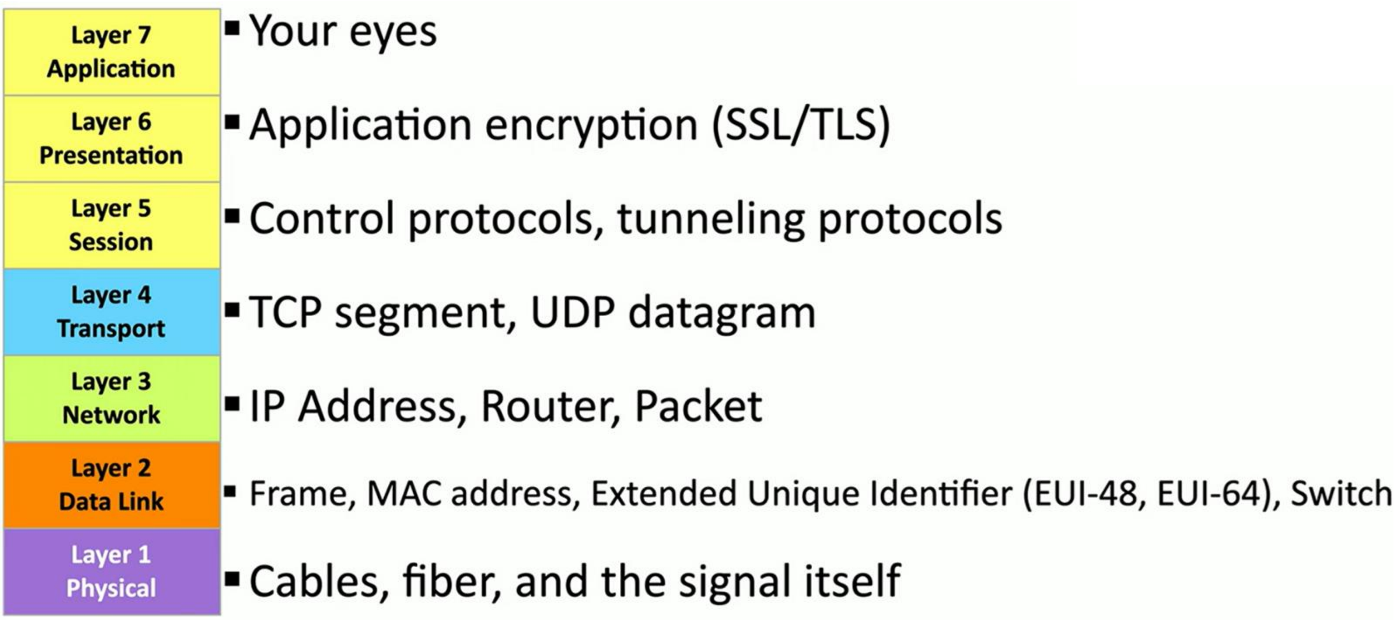
This layer acts as a translator for data to and from the application layer (layer 7). The receiving computer will also understand data sent to a computer in one format destined for in another format. For example, when you send an email, the other user may have another email client to you, but the contents of the email will still need to display the same.

Security features such as data encryption (like HTTPS when visiting a secure site) occur at this layer.

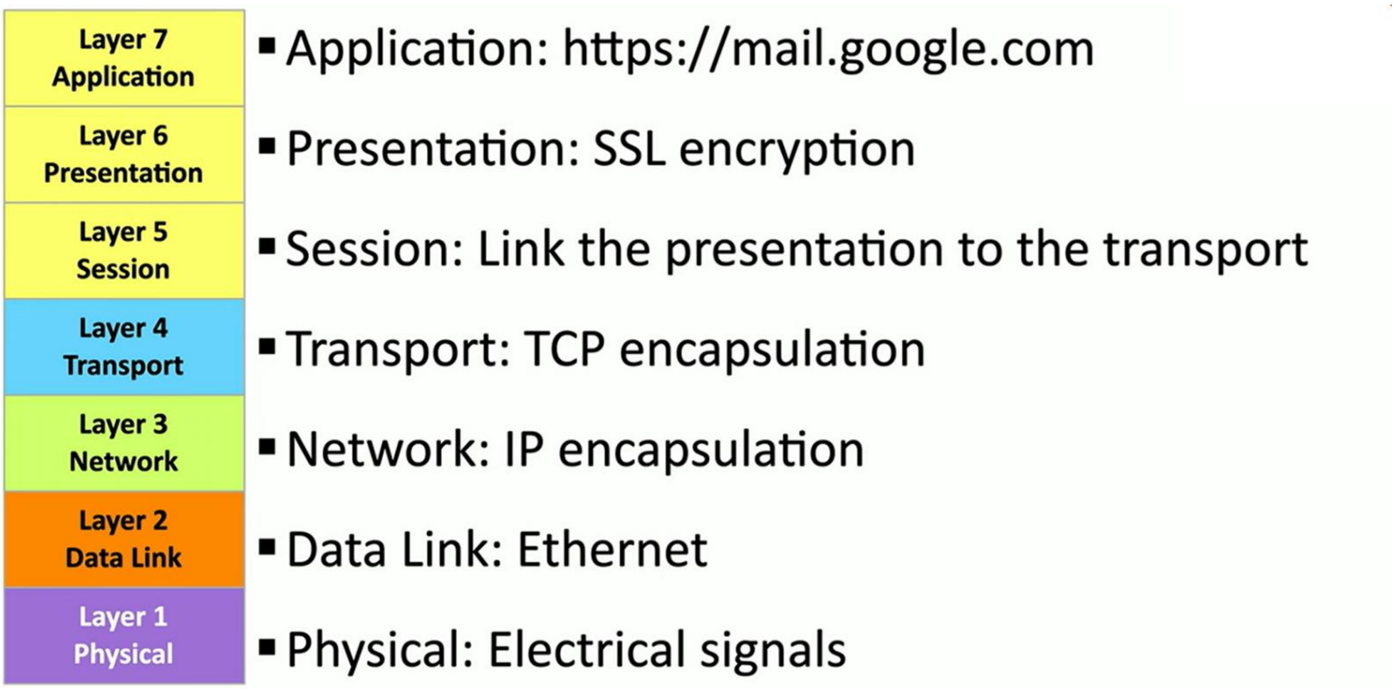
Application

The application layer of the OSI model is the layer that you will be most familiar with. This familiarity is because the application layer is the layer in which protocols and rules are in place to determine how the user should interact with data sent or received.

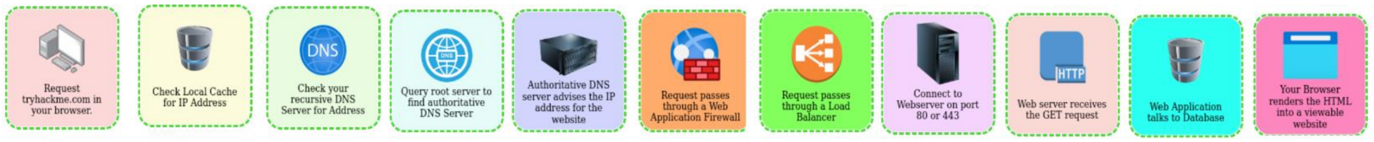
Everyday applications such as email clients, browsers, or file server browsing software such as FileZilla provide a friendly, **G**raphical **U**ser **I**nterface (**GUI**) for users to interact with data sent or received. Other protocols include **DNS** (**D**omain **N**ame **S**ystem), which is how website addresses are translated into IP addresses. The application layer is the top layer, and you might have encountered many of its protocols as you use different applications. Examples of Layer 7 protocols are HTTP, FTP, DNS, POP3, SMTP, and IMAP. Don’t worry if you are not familiar with all of them



Layer 7 Application	▪ Your eyes
Layer 6 Presentation	▪ Application encryption (SSL/TLS)
Layer 5 Session	▪ Control protocols, tunneling protocols
Layer 4 Transport	▪ TCP segment, UDP datagram
Layer 3 Network	▪ IP Address, Router, Packet
Layer 2 Data Link	▪ Frame, MAC address, Extended Unique Identifier (EUI-48, EUI-64), Switch
Layer 1 Physical	▪ Cables, fiber, and the signal itself



Layer 7 Application	▪ Application: https://mail.google.com
Layer 6 Presentation	▪ Presentation: SSL encryption
Layer 5 Session	▪ Session: Link the presentation to the transport
Layer 4 Transport	▪ Transport: TCP encapsulation
Layer 3 Network	▪ Network: IP encapsulation
Layer 2 Data Link	▪ Data Link: Ethernet
Layer 1 Physical	▪ Physical: Electrical signals


Request tryhackme.com in your browser.
Check Local Cache for IP Address
DNS
Check your recursive DNS Server for Address
DNS
Query root server to find authoritative DNS Server
Authoritative DNS server advises the IP address for the website
Request passes through a Web Application Firewall
Request passes through a Load Balancer
Connect to Webserver on port 80 or 443
HTTP
Web server receives the GET request
Web Application talks to Database
Your Browser renders the HTML into a viewable website