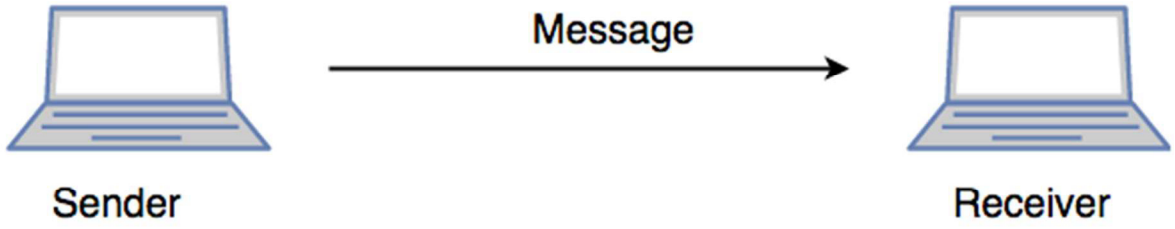
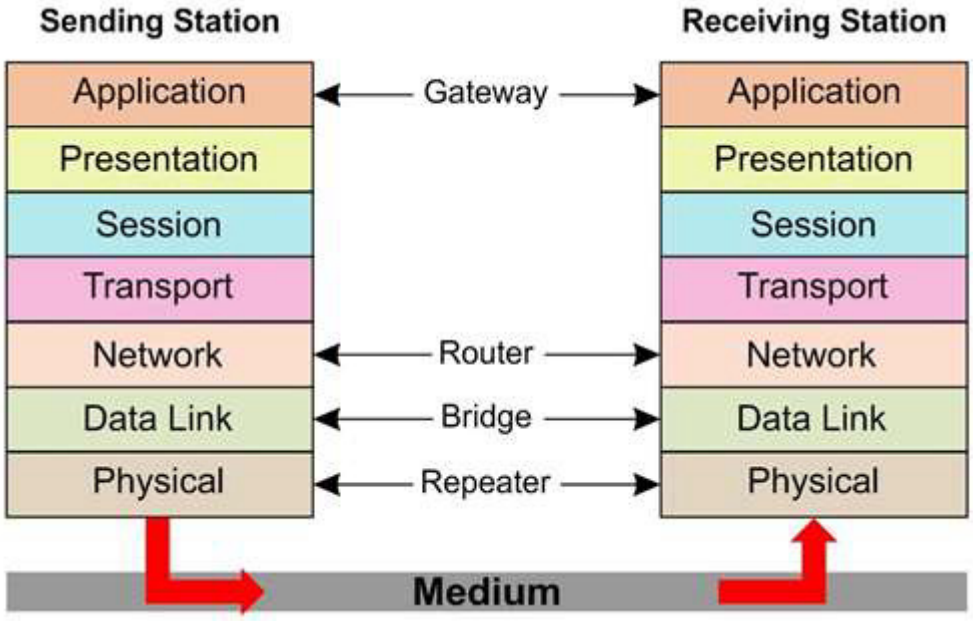
4.10 OSI Model



https://www.geeksforgeeks.org/layers-of-osi-model/

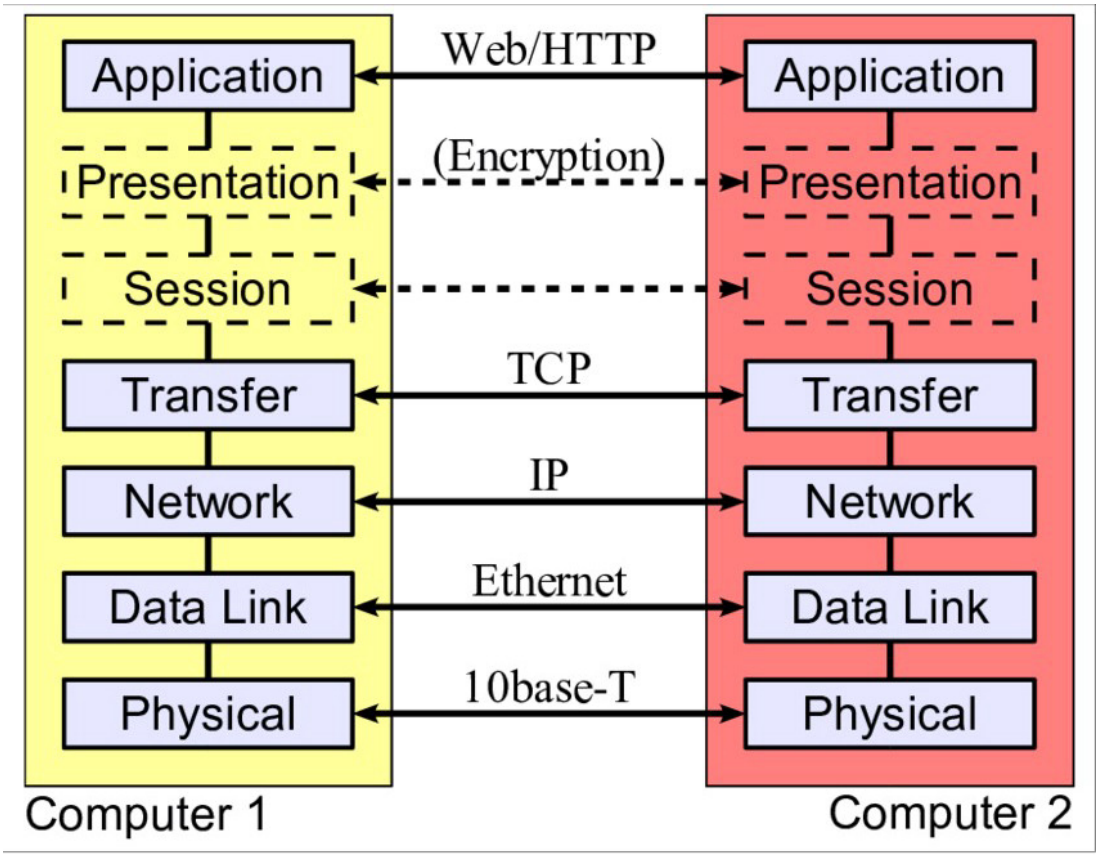
Network communications involve sending messages from one device to another. Networking software follows a strict set of rules, governed by different **protocols**, when carrying out each task. Since there are many protocols and technologies that have been developed it is difficult to get compatibility between networks. To solve this problem, a model has been developed called the **O**pen **S**ystem **I**nterconnection (**OSI**) reference model that is a description of computer network protocol design and communication.

The OSI model is divided into 7 layers through which networked transmissions pass. At each layer, different hardware is used and software implements specific network functions according to the protocols in that layer. Data from the sending application on one machine is passed vertically down through the layers from the top layer 7 to the layer 1 which is the physical cabling. The data is sent along the cable to the receiving machine which passes the data vertically up the layers to the receiving application. The different layers are able to communicate with the respective layers directly above and below them. Each layer from the sending device corresponds directly to the same layer in the destination device. The network passes data from one layer to another. As data passes through each layer a header is added on the sending side and is removed on the receiving side.



https://www.isa.org/standards-and-publications/isa-publications/intech/2011/april/understanding-ethernet-switches-and-routers-part2/

The layers communicate on a virtual level from the sending machine to the receiving machine. For example, layer 3 on the sending machine will have information for layer 3 on the receiving machine. Each layer has a definite function to perform that is defined in the OSI model. As the message passes through the layers, the data is broken up and manipulated until it is in a form that can be sent down the cable or transmission medium. The receiving machine will assemble the message and pass it up its layers until it is in a form that is ready for the receiving application.

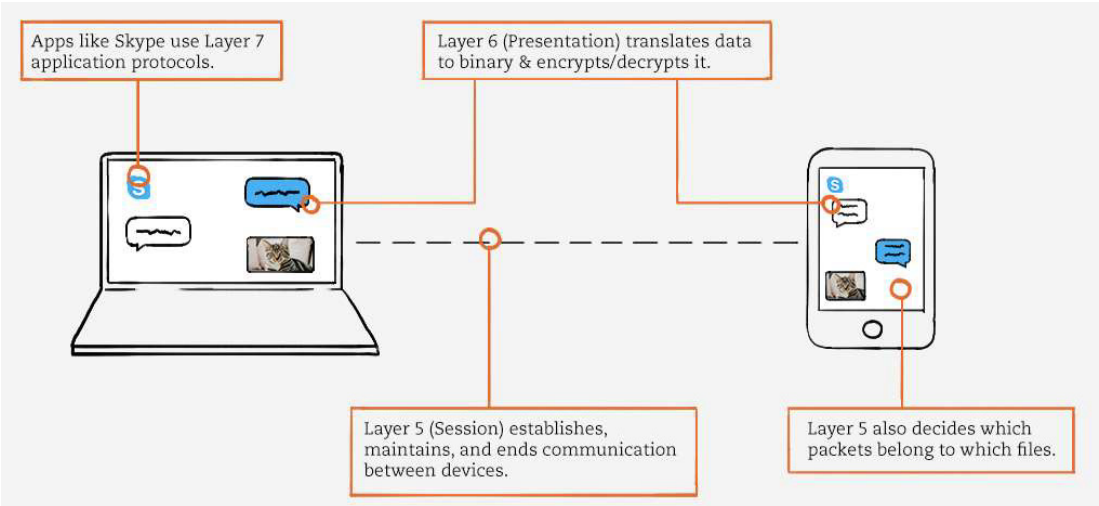


https://www.researchgate.net/figure/The-seven-layer-OSI-model-for-networking-showing-example-protocols-from-the-world-wide_fig5_290559621

4.10.1 Application, Presentation, and Session Layers

The following is adapted from: **https://www.plixer.com/blog/network-layers-explained/**

Suppose you are using Skype on a laptop to message a friend, who is also using Skype on their phone from a different network. Skype, is a network-connected application and uses Layer 7 (**Application**) protocols like Telnet. If you send your friend a picture of your cat, Skype would use the **F**ile **T**ransfer **P**rotocol (**FTP**).

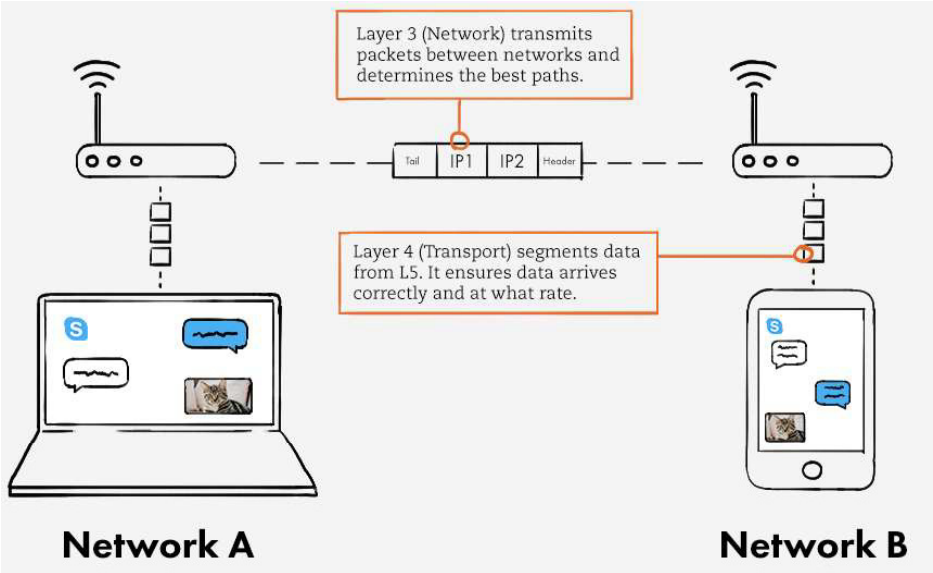


https://www.plixer.com/blog/network-layers-explained/

The Application layer will send the data to Layer 6 (**Presentation**) and the Presentation layer would translate the data into binary, and compresses it. When you send a message, the Presenatation layer encrypts that data and the corresponding Presentation layer on your friend's device decrypts the message.

Applications like Skype consist of text files and image files. When you download these files, Layer 5 (**Session**) determines which data packets belong to which files, as well as where these packets go. The Session layer also establishes, maintains, and ends communication between devices.

4.10.2 Transport Layer



https://www.plixer.com/blog/network-layers-explained/

Layer 4 (**Transport**) receives data from the Session layer (Layer 5) and breaks the data into segments. Each segment, has a source and destination **port number**, as well as a **sequence number**. The port number ensures that the segment reaches the correct application as each application has an associated port number. The sequence number ensures that the segments arrive in the correct order since the file has been broken into many segments.

The Transport layer also controls the amount of data transmitted. For example, your laptop may be able to handle a speed of 100 Mbps, whereas your friend's phone may only be able to process 10 Mbps. The transport layer can command that the server reduces the speed of the data transmission, so no data is lost when your friend receives it. But when your friend sends a message back, the server can increase the transmission rate to improve performance.

Lastly, the Transport performs error-checking. If a segment of data is missing, the Transport layer will re-send that segment.

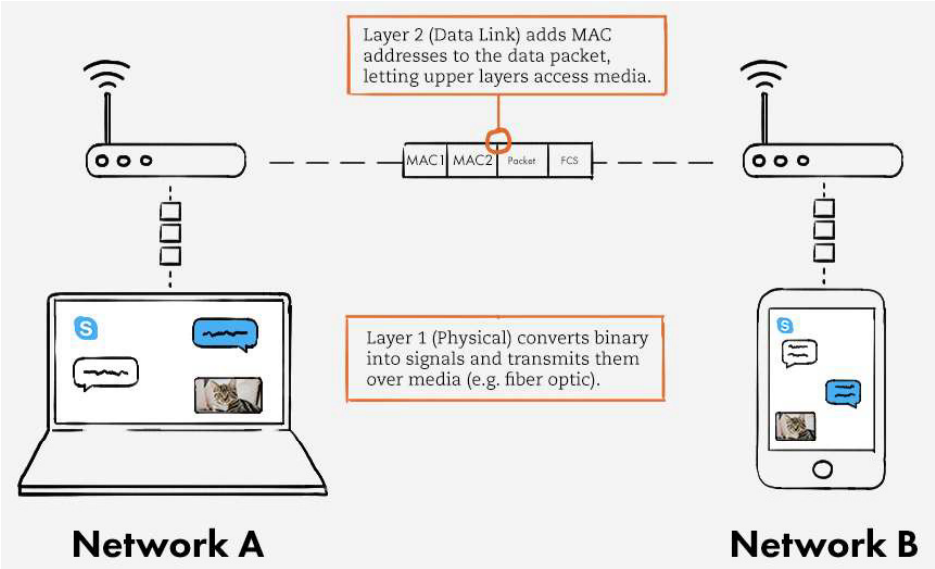
TCP and UDP are both very well-known protocols, and they exist at the Transport layer. TCP prefers data quality over speed, whereas UDP is designed for speed over data quality.

4.10.3 Network Layer

Layer 3 (**Network**) transmits data segments between networks in the form of **packets**. When you message your friend, this layer assigns source and destination **IP addresses** to the data segments. Your IP address is the source, and your friend's is the destination. The **Network** layer also determines the best paths for data delivery.

4.10.4 Data Link Layer

Layer 2 (**Data Link**) receives packets from the Network layer. Whereas the **Transport** layer performs logical addressing (IPv4, IPv6), The **Data Link** layer performs physical addressing by adding sender and receiver **MAC** addresses to the data packet to form a data unit called a **frame**. The **Data Link** layer enables frames to be transported via local media (e.g. copper wire, optical fiber, or air). This layer is embedded as software in your computer's **N**etwork **I**nterface **C**ard (**NIC**). In short, the Data Link layer allows the upper network layers to access media, and controls how data is placed and received from media.



https://www.plixer.com/blog/network-layers-explained/

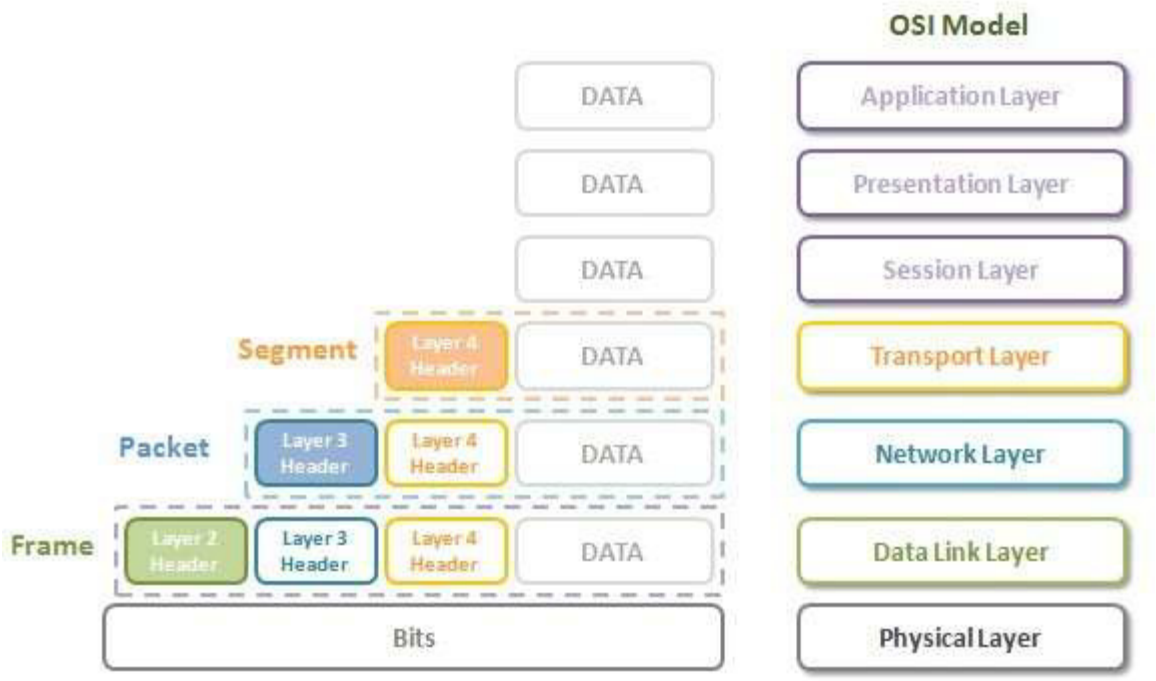
4.10.5 Physical Layer

Hardware, the things you can actually physically touch, exist at Layer 1 (**Physical**). This layer converts the binary from the upper layers into signals and transmits them over local media. These can be electrical, light, or radio signals; depending on the type of media used. When your friend receives the signals, they are translated back into binary and then into application data so your friend can see your message.

A summary of the layers is as follows:

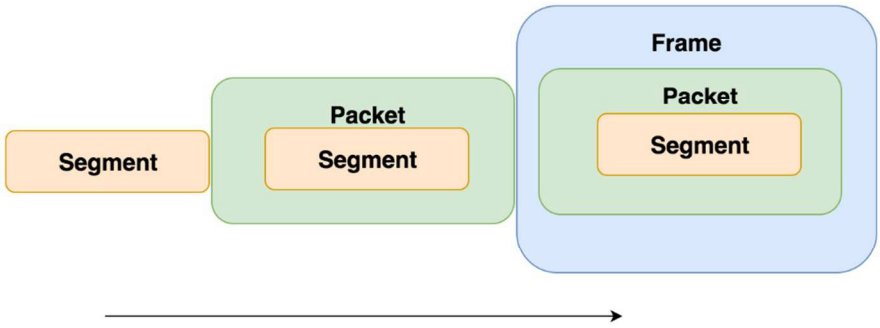
1. **Physical** (e.g. cable, RJ45)
2. **Data Link** (e.g. MAC, switches)
3. **Network** (e.g. IP, routers)
4. **Transport** (e.g. TCP, UDP, port numbers)
5. **Session** (e.g. Syn/Ack)
6. **Presentation** (e.g. encryption, ASCII, PNG, MIDI)
7. **Application** (e.g. SNMP, HTTP, FTP)

When a file or message is sent across the network, the data to be sent is usually broken into smaller more manageable parts called **segments**. Each layer in the OSI model is responsible for formatting the data as it is passed down through the layers. The **transport layer** adds a header to the data segment and passes the it to the **network layer** which add another header producing a **packet** and passes the packet to the **data link layer** which adds a header creating a **frame**.



https://ipcisco.com/lesson/osi-referance-model/

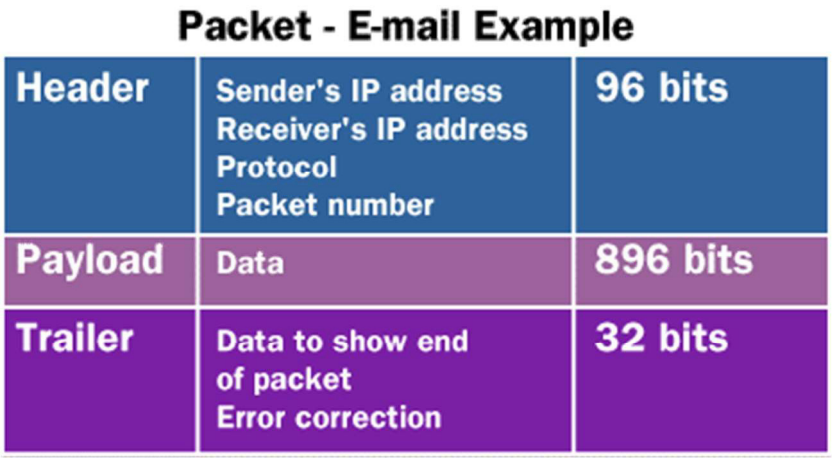
As the segment passes through the last three layers, each layer adds a header to the data. On the receiving device, these headers are removed by the corresponding layer and the data is passed up to the layer above.



https://www.slashroot.in/sites/default/files/PDU%20traveling%20through%20layers.png

4.10.6 Packet

Packets are created at Layer 3 (**Network Layer**) of the network and allow information to be exchanged between different LANs.



https://computer.howstuffworks.com/question5251.htm

Most network packets are split into three parts:

4.10.6.1 Header

The header contains instructions about the data carried by the packet. These instructions may include:

- **Packet length** - some networks have fixed-length packets, while others obtain the length of the packet from the header.
- **Synchronization** - a few bits that help the packet match up to the network
- **Packet number** - which packet this is in a sequence of packets
- **Protocol** - on networks that carry many types of information, the protocol defines the type of packet is being transmitted: e-mail, Web page, streaming video
- **Destination address** - where the packet is going
- **Source address** - where the packet came from

4.10.6.2 Payload

Also called the body or data of a packet. This is the actual data that the packet is delivering to the destination. If a packet is fixed in length, then the payload may be padded with blank information to make it the right size.

4.10.6.3 Trailer

The trailer, sometimes called the footer, usually contains a couple of bits that tell the receiving device that the end of the packet is reached. It may also have some type of error checking. The most common error checking method used in packets is **C**yclic **R**edundancy **C**heck (**CRC**). CRC works by taking the sum of all the 1s in the payload and adds them together. The result is stored as a hexadecimal value in the trailer. The receiving device adds up the 1s in the payload and compares the result to the value stored in the trailer. If the values match, the packet is good. But if the values do not match, the receiving device sends a request to the originating device to resend the packet.

Each packet's header will contain the proper protocols, the originating address (the IP address of your computer), the destination address (the IP address of the computer where you are sending the e-mail) and the packet number (1, 2, 3 or 4 since there are 4 packets). **Routers** in the network will look at the destination address in the header and compare it to their lookup table to find out where to send the packet. Once the packet arrives at its destination, your friend's computer will strip the header and trailer off each packet and reassemble the e-mail based on the numbered sequence of the packets.

https://computer.howstuffworks.com/question5251.htm
https://www.plixer.com/blog/network-layers-explained/

The **Network** layer packets allow **routers** to send and receive data across the Internet (inter-network) using the **IP addresses** that identify the network and the temporary address of the device on the network. Once inside a network, intra-network (LAN) data forwarding is handled by the **Data Link**

layer **switches** that read the **MAC address** of the frame to forward it to the destination device where the **Ethernet network card** extracts the data payload completing the process of transferring information between devices on different networks.

4.10.7 Frame

Frames are created by the Data Link Layer (Layer 2) by encapsulating the **packet** from the **Network** layer as the frame's **payload**.

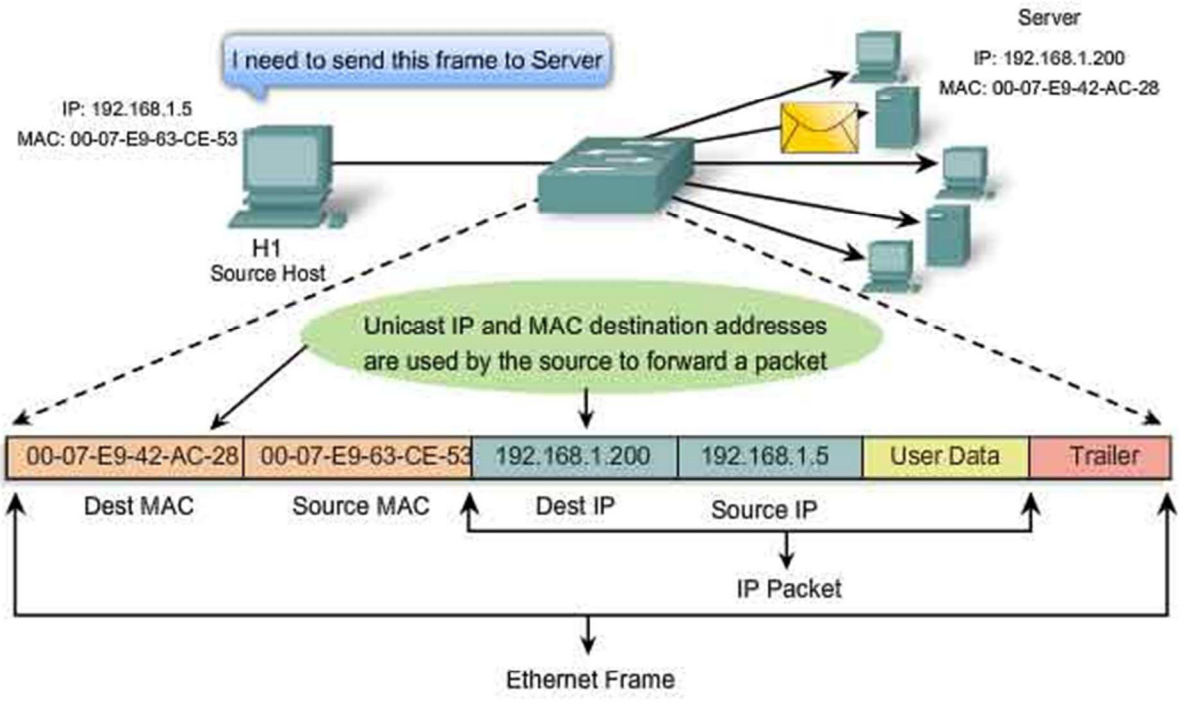
Ethernet (802.3) Frame Format

7 bytes	1 byte	6 bytes	6 bytes	2 bytes	42 to 1500 bytes	4 bytes	12 bytes
Preamble	Start of Frame Delimiter	**Destination MAC Address**	**Source MAC Address**	Type	Data (payload)	CRC	Inter-frame gap

For TCP/IP communications, the payload for a frame is a packet

The Ethernet frame adds the MAC **source** and **destination address** to deliver the **payload**, between two locations on the same network. Ethernet switches on the network check the destination address of the frame against a MAC lookup table in its memory. The lookup table tells the switch which physical port, i.e., RJ45 port, is associated with the device whose MAC address matches destination address of the frame.

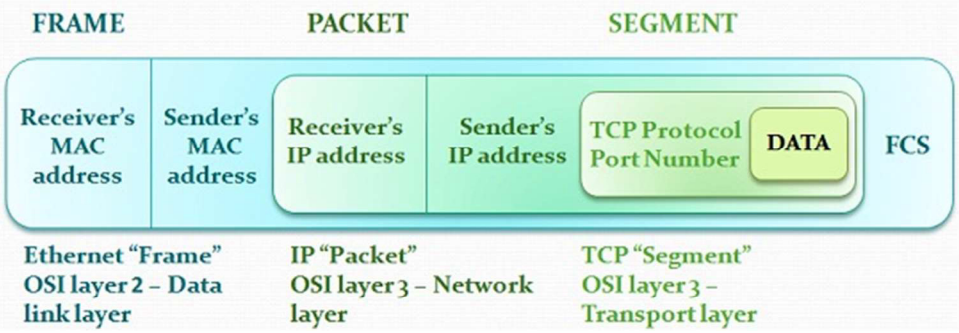
The switch will forward the frame to the physical port determined by the lookup table. If the cable is connected directly to the destination device the transmission is complete. If the cable is connected to another switch, the next switch will repeat the lookup and forward process until the frame reaches the intended destination.



http://www.highteck.net/images/230-Ethernet-frame-unicast.jpg

Switches do not deal with IP addresses only MAC addresses, the IP address is embedded in the frame's payload and a switch can only inspect the header and trailer data in a frame. A router is a more intelligent device. It deals with IP addresses by removing the MAC addresses and other frame data and extracting the frame payload which is the TCP/IP packet. Switches transfer data between the nodes of a LAN and router transfers data from a LAN to an external network like the Internet.

The MAC addresses of a network devices are unique and permanent, IP addresses are usually temporally assigned to a network device and change as the device connects to different networks. For example, the IP address of a tablet will change each time it is connected to a different Wi-Fi network.



https://techdifferences.com/wp-content/uploads/2017/08/featured-4.jpg

4.10.7.1 Frame Check Sequence

A **F**rame **C**heck **S**equence (**FCS**) is used to detect if errors have occurred in the data transmission. These are the extra bits and characters added to the end of the frames, which are checked at the destination. Again, CRC could be used as the algorithm to determine that the delivered data is correct.

4.10.8 VLAN

A **V**irtual **L**ocal **A**rea **N**etwork (**VLAN**) is a subnetwork which can group together collections of devices on in a LAN to create a new virtual LAN. Recall that a LAN is a group of nodes or devices that share a communications line or wireless link to a server within the same geographical area.

VLANs make it easy to divide a LAN into logical LANs which group similar devices based on their function or security requirements without running new cables or making major changes to the current network infrastructure. VLANs are often set up in complex LANs to re-partition devices for better traffic management.

VLANs can help improve the overall performance of a network by grouping together devices that communicate most frequently. VLANs also provide security on larger networks by allowing a higher degree of control over which devices have access to each other. VLANs tend to be flexible because they are based on logical connections, rather than physical.

4.10.8.1 VLAN Tag

VLAN tags were invented to allow LAN switches to distinguish between physical groups of LAN ports and logical groups of LAN ports. A LAN switch needs to know that "these ports belong to VLAN A" and "these ports belong to VLAN B."

The VLAN tag is a four-byte field inserted between the source MAC address and the Ethertype (or length) field in an Ethernet frame. The details of how these four bytes are made up is beyond the scope of this course.

https://www.juniper.net/documentation/en_US/junos/topics/concept/layer-2-networking-ethernet-frame-forwarding-802-1q-vlan-tag-mx-solutions.html