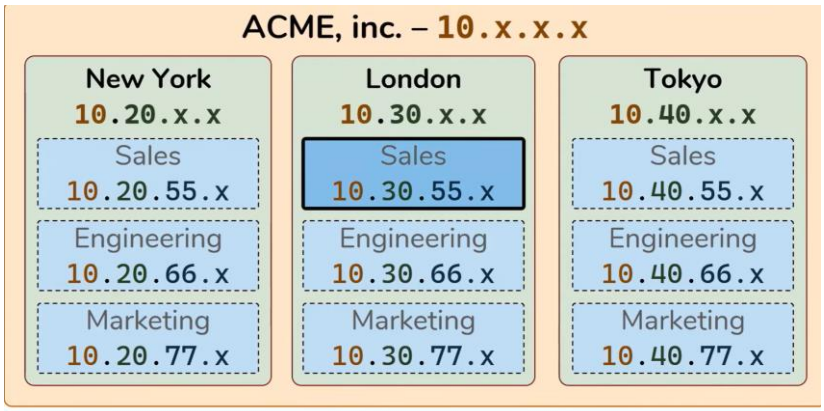
NetworkDevices:

Host:

- **Any deviece with receives and sends traffic**
- **Client requests for initiation which Server responds to. Bsp. Server= Website <- Client=PC; Server=File Server <- Client (Website)**
- **Servers = Computers with an installed Software responding to a specific request**

IP-Adresses = identity of each Host (for each Client and Server)

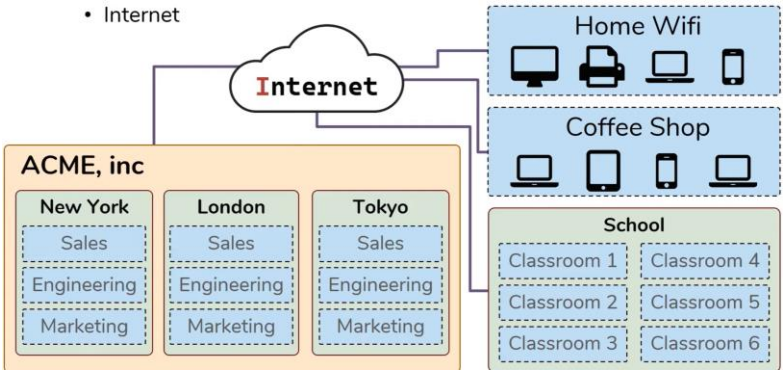
- **If a Client sends a Web requests to a website it contains which webpage it is asking for, with source and destination IP-Adresses. Src_Adr.: Clients IP Adress, Dest_Adr.: Server IP Adress**
- **When Server responds by providing Webpage it contains Src + Dest Adr too, However Src=Server, Dest=Client**
- **IP-Adresses are 32 bits (bit = 1 or 0) represented in four Octets 10001000.10001010.1011001.10011011 -> in Decimal Number Bsp. 136.22.17.98 => IP**
- **Hierachy structur: Bsp. ACM GmbH owns 10.x.x.x, office in NY 10.20.x.x and Bremen 10.30.x.x ..**



- 10.30.55.127 – Host at ACME. in London. in Sales

Network:

- **Transport traffic between Hosts**
- **Everytime two host connect there is a network between them auto. without disk, etc.**
- **Logical grouping of hosts with the similar connectivity profils (use the internet) bsp: location diffrence**
- **Network can contain othere nekwork = Sub-network/ Subnets: Bsp. NY is Subnet to ACME, inc. Network and Sales a Subnet to NY Network...**
- **Network connects to other Networks: Bsp. Homenetwork connected with Office Network from NY Sales Network to have access to Office resources (all those newtworks are connected to a central resource = Internet)**



- Internet = interconnection networks

Reaper:

- Regenerates Singles between two Hosts
- For communication across longer distances

Instead of multiply PCs habe each a connection to all of the others, all PCs have one connaction (kinda like with the internet)

Hub:

- Is a multiy-port Reaper (so regenerates Singles tfrom one Host to all the others)
- Porblem: the signle destined for one other host will en send to all other host -> Solution: Bridges

Bridges:

- Bridges sit in between Hub-connected Hosts
- Bridges learn with Hosts are on each side
- So if only Hosts on the same side communicated the signal will not pass the bridge to the other hosts

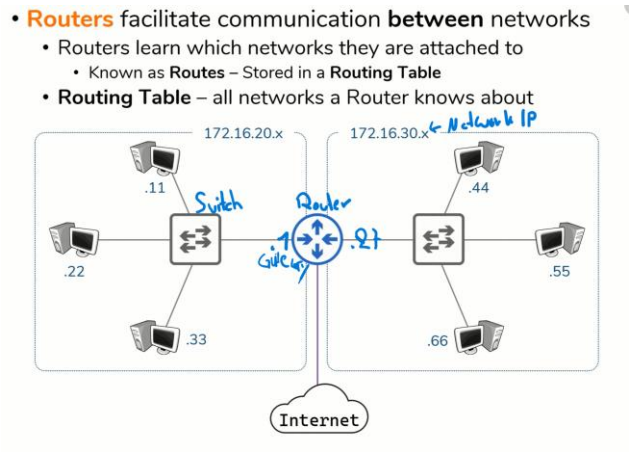
Switch

- Is a Hub and Bridge Combination
- Hub between all PCs -> multiply ports
- Learns which Host are on each port
- Facilitates the communication **within** a network (network = grouping of host which require a similar connactivity)

Hosts on a network share same IP-Adress Space -> Solution: Router

Router:

- Facilitates communication between networks
- Provide a traffic control point (security, filtering, redirecting)
- Knows which netwroks are attached also known as Routes
- Routing Table = all netwroks a Router knows
- Router give the netwrok an IP-Adress => it serves as a Gateway (each host's way out of the local network)



Routing = Process of moving data between betworks -> Router is a device whose primary purpose is Routing

Switching = Process of moving data within networks -> Switcher is a device whos primar purpose is Switching

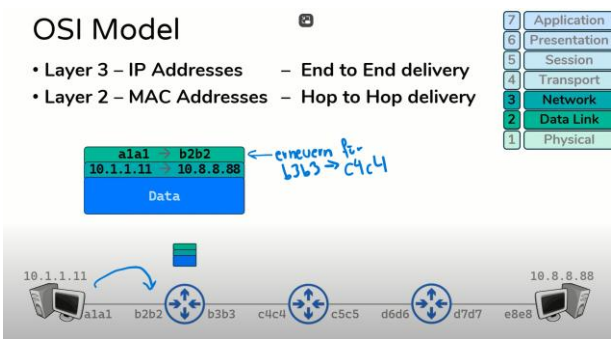
Other Network devices: Access Points, Firewall, Load Balance, Layer 3 Switches, IDS/IPS, Proxies, virtual Switches and Router (used only in the cloud) -> all of them perfom switching, routing or both

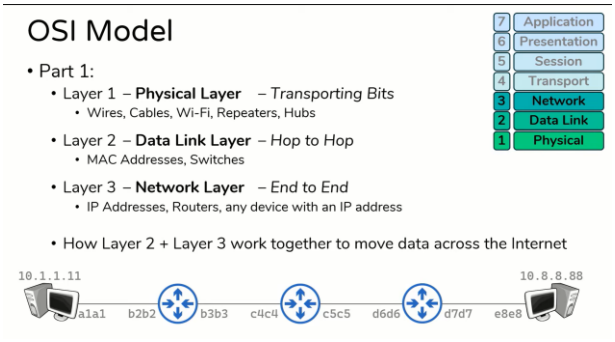
OSI Model

Each of the 7 layer serves a specific function:

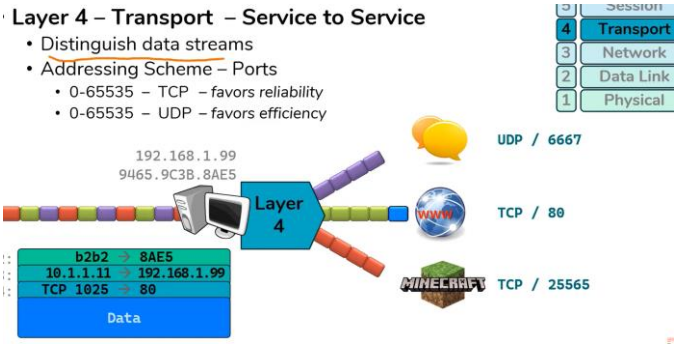
- Layer 1- Physical: PC data exists in hte form of bits (1 or 0) The physical layer transports dose bits between hosts
  Example: cabel, wi-fi (wireless internet), reapter, hub -> function to transport data
- Layer 2- Data Link: interacts with wire. Puts bit on wire and retrieves them via MAC Adresses (48 bits in 12 hexa digits)
  Exmaple: Wi-fi Access Cards, NIC (network interface card), Switches-> hope to hope delievery, help traffic move along

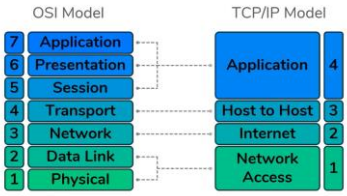
- Layer 3- Network: End-to-End delivery via IP Adresses (32 bits in 4 octets)



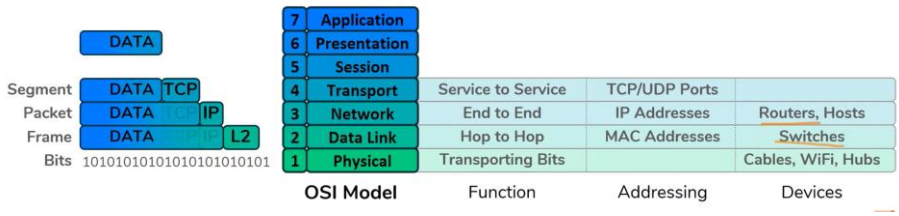


- Layer 4- Transport: Service-to-Service
  A PC has a IP- and a Mac-Adress and a web Browser and the user uses communication program (like discord) and some video game. Each of those programs sends and receives data. All data is destined to layer 3 – IP-Address End to End delivery and layer 2 –Mac-Adress for Hop-to-Hop delivery. But how does the right program geht the right data package:
  Layer 4 untercheidet data stream with own Adreses Scheme: 0-65535 TCP favor reliablity; 0-65535 UDP favor efficiency
  Server listens to requests for pre-defined Ports (HTTPS -> by default: TCP) and for each program the Server creates a random Port Adress, so the traffic is seperatet for each program
  Example: Discord has UDP/6667, Valorent TCP/80, webbrowser TCP/25565





- Layer 5,6,7:



Hosts communicated through same network

Both host have a NIC, and a MAC-Adress, IP-Adresses and a Subnet Mask

Hosts communicated through foreign network (wenn Router and Internet between)