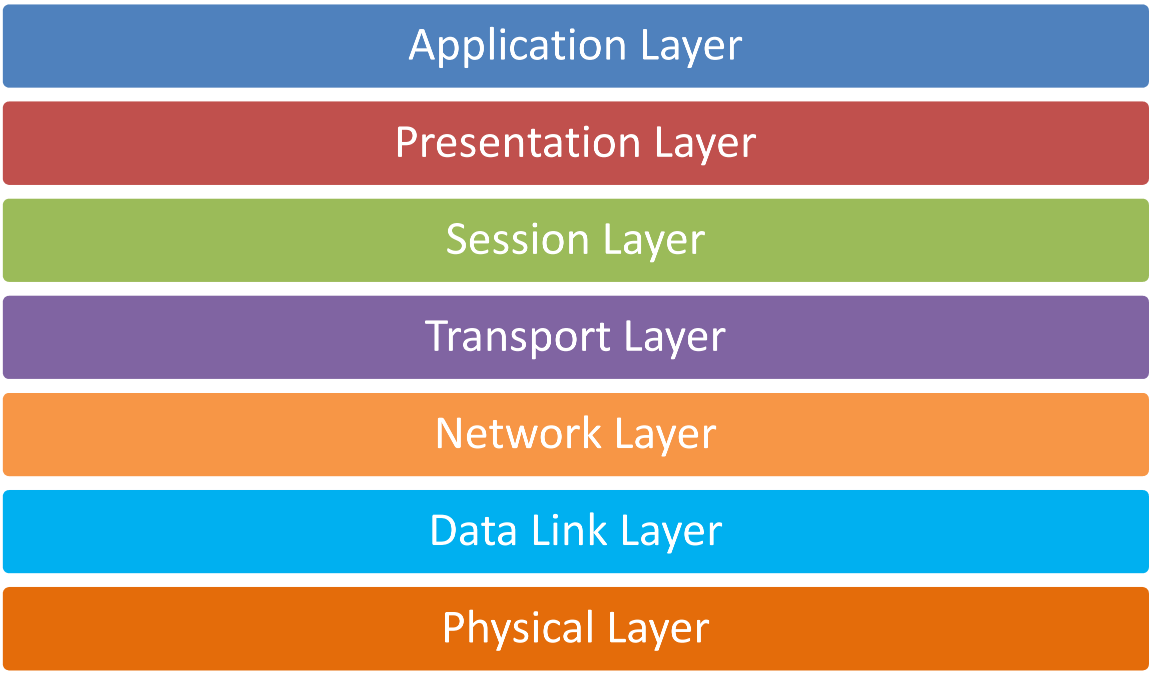
Networks – ISO OSI Model and TCP/IP Model

OSI – Open System Interconnection – model of the network system

- system is made of cooperating layers
- each layer is assigned certain functions
- each layer can directly communicate with the layer above and below only (e.g. 3rd layer may communicate with the 2nd and 4th layer only)
- each layer utilizes services of the lower level to accomplish its functions
- some layers transform data into packs for easier management (with different names – data link layers uses **frames**, network layer **packets**, transport layer **datagrams**)

OSI Layers



Application Layer

Presentation Layer

Session Layer

Transport Layer

Network Layer

Data Link Layer

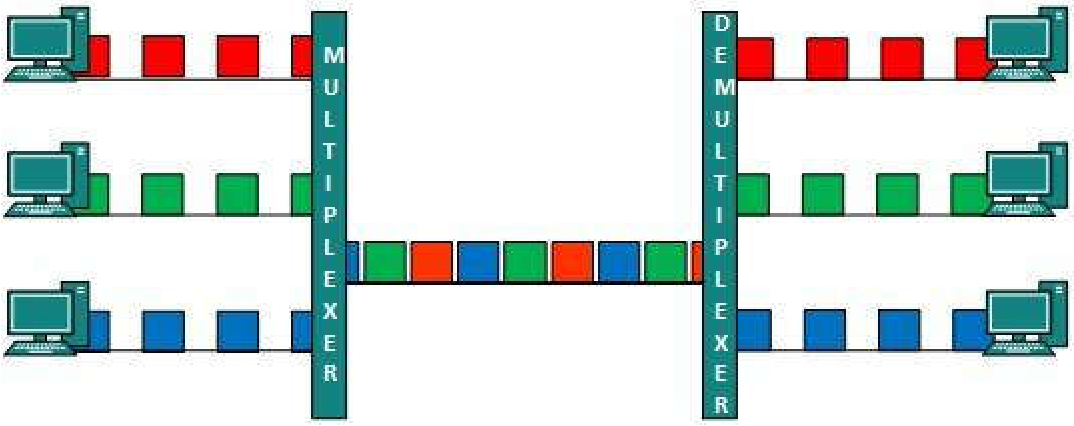
Physical Layer

Please **D**o **N**ot **T**hrow **S**ausage **P**izza **A**way (a little bit of help to memorize the order – from bottom to top).

Functions

Physical Layer

- it handles communication, which is
  - direct (between 2 directly connected devices)
  - bit-level
- PL defines physical characteristics of media and signals (voltage, impedance, physical connectors)
- Physical layer functions:
  - **Bit-by-bit** or **symbol-by-symbol** delivery.
  - Definition of the physical interface and media
    - Mechanical parameters (size, materials)
    - Electrical parameters (voltage, current, impedance, frequency)
    - Electromagnetic parameters (band, transmission output),
  - **Modulation** – a method of data carrying by the change of periodic waveform (e.g. Manchester code in Ethernet).
  - **Multiplexing** – a combination of multiple signals into one (its opposite → **de-multiplexing**) – e.g. using frequencies (FDM) or quantum of time (TDM).



  - **Carrier sense and collision detection** (e.g. used in CSMA/CD).
  - Some further functions (circuit-switching, bit synchronization etc.).

Data Link Layer

- It handles communication, which is
  - direct (between 2 directly connected devices)
  - frame oriented – data is portioned into **frames** (groups of bits), which are either sent to physical layer (transmission) or taken from the physical layer (receiving)
- On this level are defined **MAC addresses** – to identify the communicating devices

- Consequence: MAC address cannot be used as identification for communication between devices, which are not directly connected – when the data is sent, MAC addresses must be constantly changed → data link layer cannot handle communication **between** networks.

Network Layer

- Network layer is responsible for the **routing of packets** through the **entire network** – it makes decision where to send incoming data
- Routing is done by logical addresses – **IP addresses**, which are kept in packets during the communication (unlike MAC addresses in frames)
- Packets = groups of bits

Transport Layer

- It is responsible for providing **reliable** connection, if necessary
- It takes continuous stream of data from the session layer and turns them into **datagrams** (when transmitting)
- 2 important protocols are defined here
  - **TCP** – connection oriented – it guarantees the delivery of data using acknowledgements – each datagram must be acknowledged as being correctly received (if some data is not correct or not delivered, TCP protocol arranges its re-transmission)
  - **UDP** – connectionless protocol – simply sends data and does not care for correct delivery (e.g. used in multimedia streaming – video, audio – where some missing data can be accepted)

Session Layer

- Works with sessions – connections between two sides – the goal is to establish, maintain, and finally tear down the connection.
- *Example*: a connection to Facebook established over a browser. Even if the network is changed (i.e. different IP address, different topology etc.), the connection is automatically re-established using the session identifier, which is stored on both the server and browser (as a cookie).

Presentation Layer

- Converts data for further processing, e.g
  - Compression / decompression do decrease the size of transmitted data
  - Encryption/decryption to secure the communication
  - Conversion of character sets – different systems (Windows, MacOS, Linux, Android …) use different sets of characters and data – presentation layer hides them

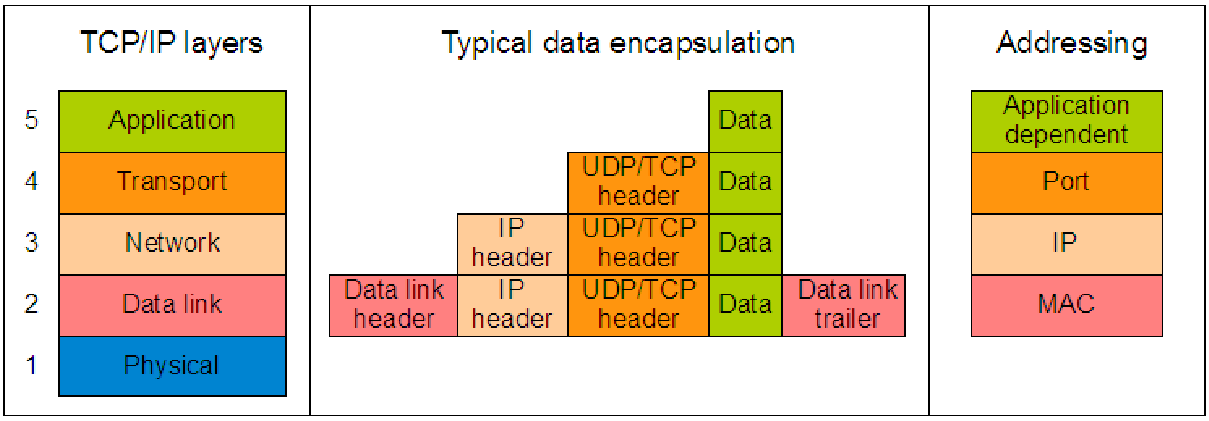
Application Layer

- Implemented by various applications, which utilize various protocols, e.g.
  - HTTP – transfer of web pages,
  - FTP – transport of files,
  - POP3/IMAP – email delivery,
  - BitTorrent – exchange of files over the P2P (peer-to-peer) network,
  - DHCP – automatic setting of network settings for connected devices (IP address, subnet mask, gateway, DNS)
  - SSH – remote connection to CLI (command line interface) of a computer

Data Encapsulation in Networks

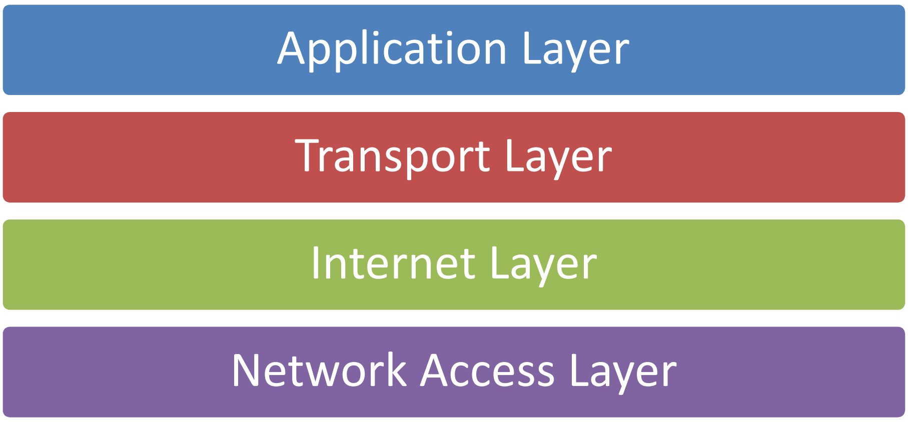
Encapsulation is a process, in which data unit (**datagram**) from a higher layer is placed in the data area of the unit in the current layer. This happens, when the data is being prepared for the transport and passes the layers from the top one towards the bottom one (physical layer).

The opposite process, when the data is received, is called **decapsulation**.



TCP/IP Model

- Another layered model – the closest one to networking used on internet-based networks (*carefully, Internet and internet are two different things → Internet is the world-wide network, while internet is a networking technology based on the TCP/IP protocols)*



Application Layer

Transport Layer

Internet Layer

Network Access Layer

Network Access Layer

- Similar to Physical Layer in OSI model – defines physical characteristics of the signals and media
- Protocols: Ethernet, **WiFi: 802.11b/g/n, ATM**

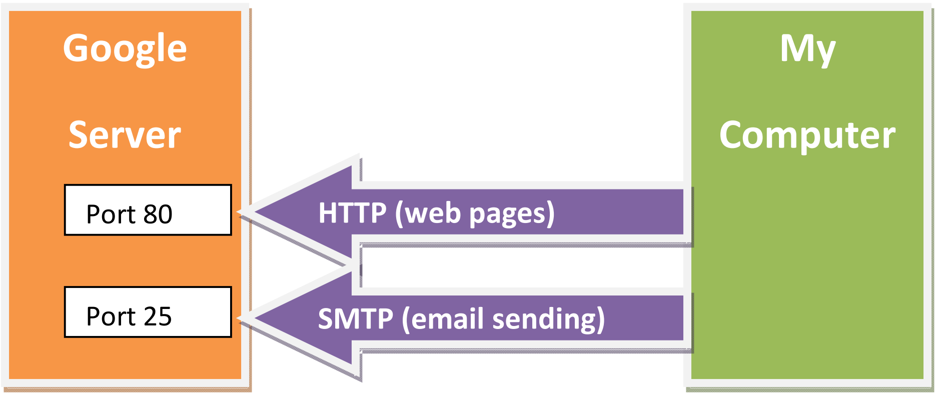
Internet Layer

- Internet Layer is responsible for
    - **packetization** – transformation of the data stream to packets,
    - **addressing**, and
    - **routing** of data on the network from one computer to another – i.e. it defines methods how to direct packets between networks.
- Protocols: **IP, ICMP** (you might be familiar with *ping* or *tracert* – those are utilities, which use the ICMP protocol)

Transport Layer

- Very similar to Transport Layer in OSI model
- It provides the means for the transport of data segments across the Internet Layer. The Transport Layer is concerned with **end-to-end** (host-to-host) communication (it does not try to solve the routing – that is the responsibility of the Internet Layer).
- Here are defined **ports**
    - There are usually multiple applications on a computer, which want to communicate over the network. The computer has one IP address (usually), which cannot be used for decision, which application should get the data. That’s why the ports are defined – they are just numbers (16-bit integers), which identify virtual channels on the computer. Each port can be assigned to one application, which *listens* then on the port.

    - Example (simplified) – I want to look for some webpages using Google and at the same time I want send couple of emails using Gmail – that means I want to contact the same server, but 2 different programs running on that server – to make a difference the transport layer puts appropriate port numbers to datagrams (80 for HTTP, 25 for SMTP; other common ports are 53 for DNS, 443 for HTTPS)



- Main protocols: TCP, UDP

How to Show Open Connections

Open **cmd**, type and press Enter:

```
netstat -a
```

Application Layer

- It comprises functions of session, presentation, and application layers from OSI
- It forms the interface for users