Recap from Friday: OLED display

- OLED (like also TFT displays) are modules composed of
  - The Matrix of OLEDs organised in rows and columns
  - A controller chip which contains the image to be displayed in a memory

- The controller chips has the following functions:
  - Interface to the user via a standard bus (I2C, SPI, or parallel bus)
  - Allows the user to write pixel information into the RAM
  - Controls or “drives” the display matrix:
    - The matrix is displayed by scanning fastly through all pixels and either switch then on or off. Each pixel has a capacitor which makes the pixel light up for longer than the time it is switched on (the principle is similar to old Cathode Ray Tube displays but the physics behind is different there (a electron beam scanning a screen which has some afterglow)
    - This involves the setup of various voltages and clocks. They are specific to the OLED matrix. Usually voltages higher than the supply voltage are needed to operate the OLEDs. These are generated in DC/DC converters.

Recap from Monday

- Operation of the OLED display
  - We saw the OLED display is being operated differently than the usual “register based” I2C devices:
    - A Control byte is transferred at the beginning of each transaction
    - It defines 2 things:
      - If the following bytes are to be interpreted as “Command bytes” or as Pixel bytes going into the pixel memory
      - If after the next byte another control byte is transferred or if only data bytes follow in the ongoing transaction

OLED module

- For convenient usage: Framebuffer
  - A framebuffer is a copy of the Pixel map which is maintained in the software.
  - The image is constructed in this buffer, then the buffer is transferred in one go to the OLED module
    - This can be fast and efficient since the controller supports long I2C block transfers where 128 bytes can be transferred in one go
  - For the framebuffer there exist libraries with a lot of convenient methods (i.e. draw text)

Recap : Use display in landscape mode

```
from machine import I2C, Pin

import framebuf

from SH1107_OLED import OLED

OLED_ADR = const(0x3C)   # decimal: 60

i2c = I2C( 0,scl=Pin(32),sda=Pin(33),freq=100000)

oled = OLED(i2c, OLED_ADR)

oled.init()

oled.setLandscape(True)

fb = oled.getFramebuffer()

fb.fill(0x0)

fb.text( " PADOVA", 0,10,1 )

fb.ellipse( 32, 13, 31, 10, 1)

oled.copyFramebuf( )
```

Microcontrollers

Part IV

Basic Networking

(Christoph.Schwick@cern.ch)

Networking

- Networking allows to exchange information between computing devices
  - Receive input for the application
    - Measurements from a remote sensor
    - Data from a remote database
    - Data from a remote service on the internet
  - Provide information for other applications or devices
    - Essentially the above from the perspective of the other communicating party

- For this to happen data needs to be packaged and provided with some additional information telling the networking software and hardware how and where to send the data.

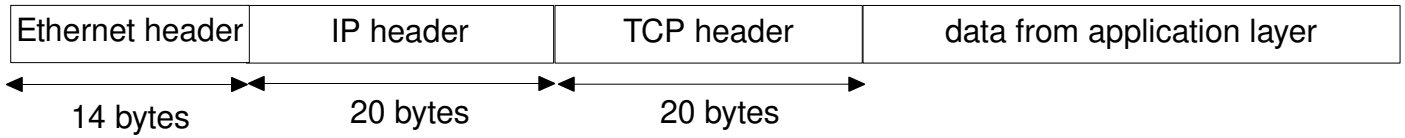
  A set of standard network “protocols” have been defined to accomplish this

Networking layers

- Data is sent around the internet is so called packets
- Packets have headers (some meta information necessary to transport the packet over the internet) and “payload” (the data you are really interested in)
- We have different “layers” of networking protocols with different purposes
  - Each layer has some well defined functional purpose
  - Each layer has its header information and its protocol
  - The headers are chained one after the other at the start of the packet, before the packet is sent to the network hardware

Typical Networking layers

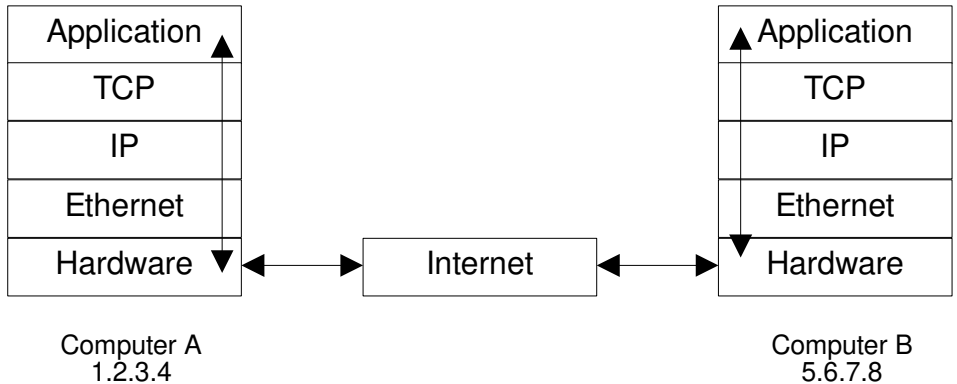
A packet on the network.



The headers of the layers are chained one after the other

An often used diagram to symbolize the network layers and their interpretation…

...a bit confusing according to me...

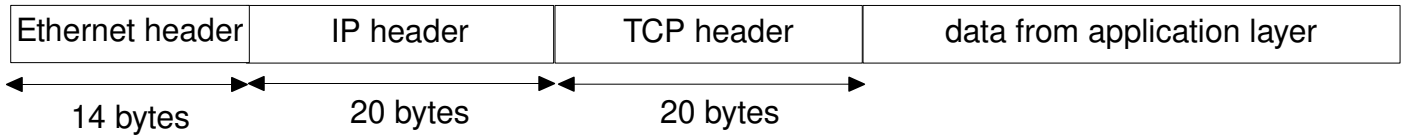


Different Layers have different purposes and are interpreted at different moments of the packets "journey" through the network

The IP header is interpreted by routers on the Internet (not really clear in this image)

Typical Networking layers

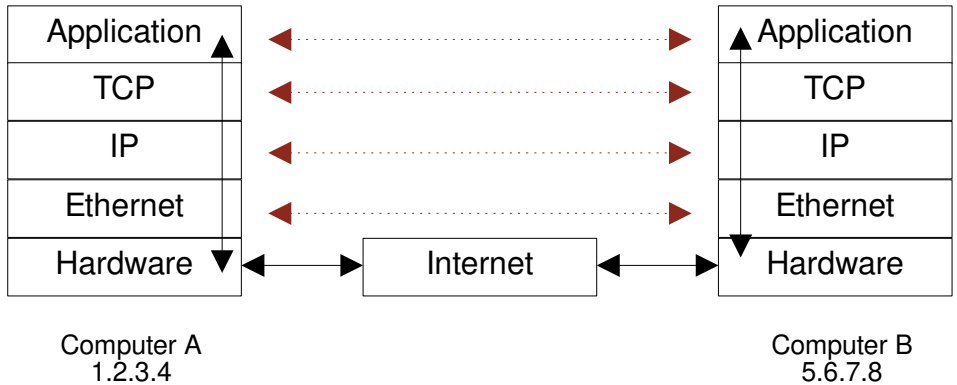
A packet on the network.



The headers of the layers are chained one after the other

An often used diagram to symbolize the network layers and their interpretation…

...a bit confusing according to me...



Different Layers have different purposes and are interpreted at different moments of the packets "journey" through the network

The IP header is interpreted by routers on the Internet (not really clear in this image)

In full detail : Application layer are three layers

- The 7 layer OSI model (from Wikipedia)
  - ATTENTION: the OSI model is an **abstract model** and does **not always fit** well to what is really ongoing in a particular network protocol suite.

OSI model

Layer			Protocol data unit (PDU)	Function[26]
Host layers	7	Application	Data	High-level protocols such as for resource sharing or remote file access, e.g. HTTP.
	6	Presentation		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5	Session		Managing communication sessions, i.e., continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4	Transport	Segment, Datagram	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media layers	3	Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2	Data link	Frame	Transmission of data frames between two nodes connected by a physical layer
	1	Physical	Bit, Symbol	Transmission and reception of raw bit streams over a physical medium

Networking layers (2)

- "Physical" or "Hardware Layer" (e.g. copper, fiber, wireless)
  - Defines how the bits are sent from sender to receiver over a physical medium
  - Essentially the electronics of the network card
    - One or more cables
    - An optical fibre
    - Wireless communication (modulated electromagnetic waves)
  - The physical layer might change during the journey of the packet
    - e.g. Your computer uses a wireless WIFI card to your home-router.
    - Your printer might be connected to your home network with an Ethernet Cable. Hence when you print : WIFI → Ethernet
    - The Router uses cables to connect to your ISP via the telephone line (ADSL)
    - Your ISP probably use fibres at some point
- "Data Link Layer" :(e.g. Ethernet, PPP, Zigbee)
  - Defines how date frames are transported between connected nodes on a specific physical media
  - Example is Ethernet (others are PPP (used by old modems) and Zigbee)
- "Network Layer": Internet Protocol (IP)
  - This protocol describes how network packets are transported to the destination in the internet.
  - Unique addresses are included in the IP protocol for source and destination of the packet.
  - IP addresses are 4 byte numbers. We write them like 137.138.34.78 (every byte is written as a decimal number, separated by points)

Networking layers (3)

- "Transport layer": Often TCP or UDP
  - Connection oriented:
    - TCP (e.g. used in HTTP (web browsing)
      - a logical connection has to be established before communication is possible
      - Reliable : data is guaranteed to arrive (it is retransmit if not acknowledged in a given time)
      - Data is guaranteed to arrive at the receiver in the same order as it is sent.
  - Connection less:
    - UDP (used for media streaming: less overhead)
      - No connection, no guarantee of data delivery, no guarantee of sequence
  - Segmentation and reassembly of data
  - This layer decides which application receives an incoming data packet on a computer.
    - To make this decision the TCP & UDP use 16 bit numbers: **the port number**
    - Applications "listen" to incoming packets on specific ports
  - Examples:
    - Port 80 for HTTP
    - Port 443 for HTTPS
    - Port 20 and 21 for FTP
    - Port 110 for POP3 (email clients (not encrypted))

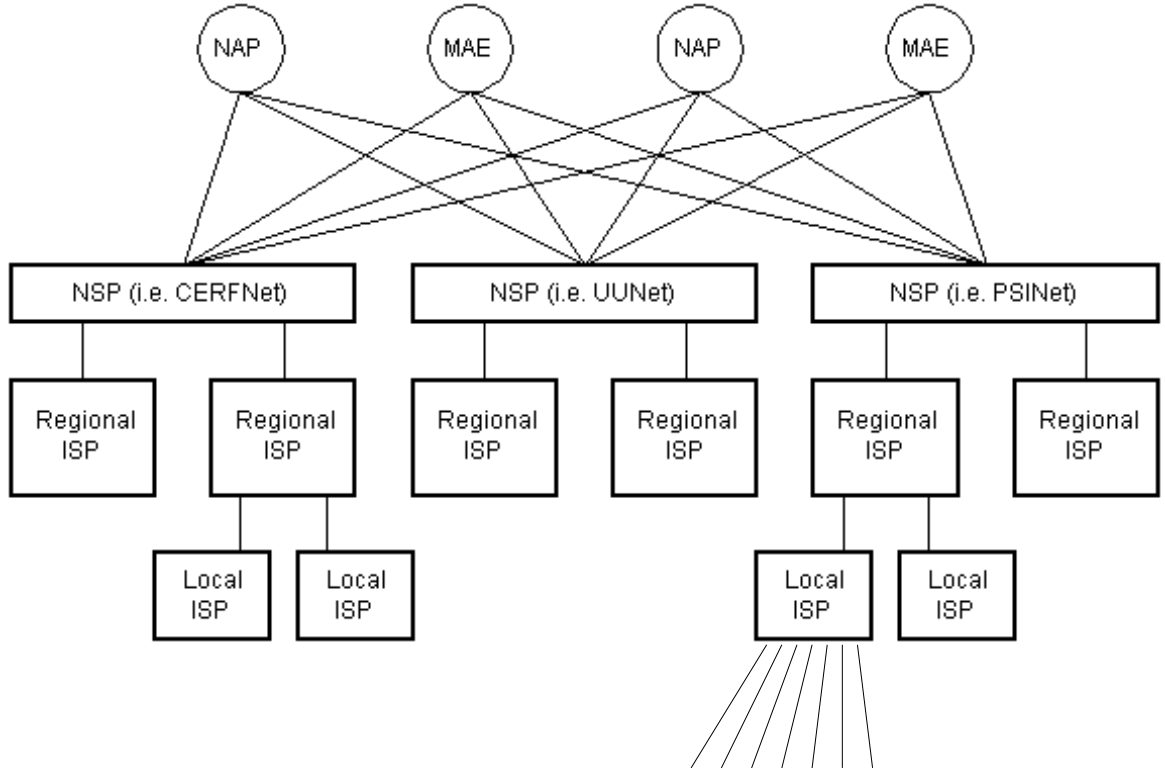
Networking layers (4)

- Session layer (NetBIOS in windows, RPC : Remote Procedure Calls)
  - Handles multiple data exchanges forth an back; Manages so called sessions
- Presentation layer (Example SSL or TLS (also used in HTTPS))
  - Handles encoding, encryption and data compression (e,g,encoding of special characters in html; https encryption)
- Application layer (HTTP, FTP, SSH, SMTP, POP3)
  - This is the protocol used by the application. Examples are
    - HTTP or HTTPS for web browsers
    - POP3 for email clients (receiving mails) and SMTP for sending mails
    - FTP for file transfer programs

Network : routing

Extremely simplified internet architecture

NAP	Network Access Point - connecting different NSPs (public)
MAE	Metropolitan Area Exchangers: - connecting different NSPs (commercial)
NSP	Network Service Provider - on top of large subnets
ISP	Internet Service Provider - on top of smaller subnets



Networking: Routing

- The process of guiding a network packet through different sections of the internet to it's destination is called "routing"
- On the internet the **IP protocol** together with the hardware components in the internet perform this job.
- The internet is divided into a **hierarchy** network chunks (**subnets**) which belong to groups of network addresses
  - The subnets are organised in a **tree structure**
  - A subnet is a range of internet addresses where the leading bits are fixed.
  - Example : 137.138.x.y is a subnet which is reserved for the CERN laboratory.
- **Routers** interconnect the subnets. They are the connection points in the tree of subnets
  - A router is connected to a number of subnets. And they have a link to one (or some) higher level routers in the hierarchy.
  - They know the address ranges of the various subnets they are connected to
  - If a packet is received which does not belong to one of those subnets it is forwarded to the higher level (the so called "**Gateway**")
  - At the top of the tree there are the "Network Service Providers" which are interconnected via Network Access Points (NSP) or Metropolian Area Exchanger (MAE) (essentially big router-like objects)
- **Network Switches** connect devices in the SAME subnet
  - Usually they work on the Data Link Layer
  - Ethernet Network switches use the MAC addresses of Ethernet to determine where to transport data packets.

    Very often switches also have routing capabilities with an "uplink" (i.e. a connection to the "outside world"). E.g. your wireless router at home.

Private subnets and NAT

- Private Subnets
  - There are some subnets reserved for private usage e.g. 192.168.0.0/16 or 172.16.0.0/12 or 10.0.0.0/8
    - The number after the '/' defines the number of leading bits in the address which are frozen
    - Hence 192.168.0.0/16 means all 65536 addresses which start with 192.168.
  - Addresses from private subnets are not known outside of the private network
    - Hence many private networks 192.168.0.0 can co-exist in the world. They are isolated from each other.
  - These subnets are used by your Home WIFI router: the router itself has an IP address from your ISP, however, inside your home network you have more than one device (multiple laptops and phones in the family, modern TVs, …) all these devices get an IP address within the private subnet (the router distributes the addresses and memorises them).
  - The networking to the outside world requires a bit of work from the router. If your grandmother clicks on a link in her browser, the corresponding page needs to be sent to your grandmothers computer and not to yours. For this the router memorises the source address of the outgoing request. Then it replaces the source address with the IP address of the router (provided by the ISP) and the answer comes back to the router. The router recognises that the incoming packet is part of the network traffic initiated by the grandmother (there are identification fields in the headers which allow to do this) and replaces the destination address (which was the IP address of the router) with the private network address of your grandmothers laptop. This method is called **NAT (Network Address Translation)** and is used by all home networks to work with multiple device which want to access the internet.

DNS (Domain Name Service)

- Memorising IP numbers is tough for human beings
  - This is why logical names have been invented:
    - cern.ch
    - google.com
    - infn.it
- These “domain names” correspond to internet numbers.
  - Before a data backet is sent to a computer in a domain (e.g. mycomputer.cern.ch) it has to be translated into a IP number.
  - This is done by the Domain Name Service (DNS)
  - Every computer knows the address of a DNS and can ask for the IP address of a logical address.
    - (In linux you can use the “dig” command to query a dns)
  - Similar to the routers on the internet, DNS are organised in a tree structure
    - If a DNS does not know the answer to a request it sends it further up in the hierarchy
    - Eventually the request travels down another branch of the tree until the answer is found and sent back

DHCP (Dynamic Host Configuration Protocol)

- Who distributes IP addresses ?
  - Companies and big organizations buy IP addresses (and domain names)
  - When you get Internet from your ISP an IP address is given to you by your ISP when you switch on your router at home
  - Similarly your laptop gets an IP address over WIFI *in the private network* when you switch it on. This IP address comes from your WIFI router:

- DHCP server and protocol
  - This distribution of IP addresses is done by “DHCP servers” (your wifi router has one).
  - The Server is configured to distribute IP addresses in a specific range (e.g. 192.168.1.10 to 192.168.1.200 for max 190 devices)
  - Every device asks for a IP address on the network at power up. The DHCP server memorises which IP addresses are still available and distributes them to the devices on the network.
    - The IP address is valid for a specific time only : the lease time (e.g. 11 hours). Then they have to be re-asked
    - Normally the same device gets always the same IP in the home network since the router memorises the ID of the device which is contained in the request of the device
  - At the same time the DCHP server gives the computer the IP address of the DNS.

Useful tools for networking (Linux)

- ifconfig
- iwconfig
- ip (powerful but very complex)
- tcpdump
- route
- dig
- tracepath or traceroute
- ping
- netstat
- **wireshark** (extremely useful for learning what goes on on the network !!! with GUI)

Exercises

- We create a dedicated wireless network for our microcontrollers:
  - SSID : “student14”
  - Password : “$unilab1”

- When writing the function to connect to the network do not hardcode these parameters but use a configuration file.
  - The configuration file should have json format:

```
{
    "ssid"              : "student14",
    "password"          : "$unilab1"
}
```

  - The code for a class to read such a config file and to then read the values of single parameters can be found in the Code examples (config utility)
  - DO NOT FORGET to upload the python file with this class and the configuration file itself to the microcontroller before you use it.
  - Be sure you understand how the code works (ask in case of questions!)