The Document

João Morais

November 1, 2019

Abstract

This will be the document where I write everything I know about what I learn during my studies. I hope it serves me in the future as a "go to" document when I need to remind something.

Introduction

This document will focus on definitions, understanding concepts and transmitting intuitions/perspectives.

Each chapter will have the same structure. First the definitions and logic connections between everything with references to derivations and demonstrations that will be in attach at the end of the document. Second an overview of the derivations and how formula connect together to make a quick & easy way of finding the connections for hurry times. And third, simply one or two pages with the most important formulas from that chapter.

The first chapter will be about antennas. The second about microwaves. And the ones after that will consist of applications such as Microwave Links/ Hertzian Beams, Satellites and Radar. Maybe I'll join in a chapter about Probabilistic/Statistic Detection and Estimation of signals and one about some fundamentals of communications.

Please be aware that this is a document always at work. I'll write right here when I think a certain chapter is closed, otherwise it might be target of modifications.

Finally, there's a resource folder that I'll be refferencing frequently. It has many of my notes hand written, books, slides. Things I thought to be interesting or important. Probably all books mentioned in any section will be there.

This folder can be accessed through my Shared Drive Link

Enjoy

Contents

2 Telecommunication Networks - Transport Networks

From the courses Internet Networks and Services (RSI in portuguese) and Telecommunication Networks (RTel in pt) I've had an insight about how the whole network is multiplexed into optic fibers and many other interesting topics such as the triple play services and a bunch of protocols that are used in today's world to make everything communicate with everything. Therefore, I propose to write a sum up of the slides and bibliography of RSI and RTel in this section. I'll mainly give importance to RTel since it is what I'm studying at the moment, but I hope to go through the slides of RSI as well.

1. Introduction
   (2 lessons)
2. Fundamentals of networks
   (7 lessons)
3. Ethernet and data centre networks
   (5 lessons)
4. SDH transport networks
   (4 lessons)
5. Optical transport networks
   (4 lessons)
6. Access networks
   (3 lessons)

1. The Internet
2. Quality of Service on the Internet
3. IP Network Models
4. Next Generation Networks
5. The Telephony Network
6. Technologies for data transport
7. MPLS - Multi-Protocol Label Switching

2.1 Introduction

Definition *Telecommunications* : is the transmission of information at a distance through the use of electromagnetic signals.

Definition *Telecom. Network* : collection of nodes and links with the purpose of interchanging these signals in order to have an information flow.

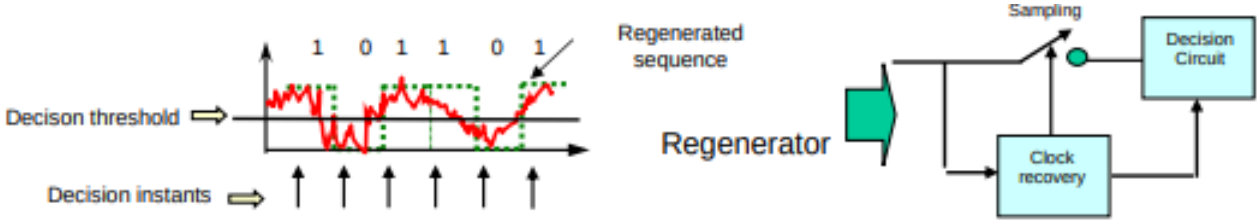
These Telecommunication Networks can be public, owned by Telecom. / Network Operators that use that network to provide services to the general public, or can be private, used by a company to connect infrastructures. Many of these private networks also rely on leased links by public networks.

There are mainly 3 layers in a network: the backbone or core, the metropolitan and the access layer. As expected, the access layer collects the traffic, connections to homes, offices, everywhere the internet is required. The metropolitan area connects different parts of the city that use that network, typically with a ring (made out of optic fiber). The core is the most extensive layer, with a mesh of nodes and very extensive links that connect cities of the whole world. Hundreds or thousands of km's is not atypical.

What makes possible to communicate with everyone connected to the internet is that the networks of both operators are also connected.

As a public service, the public networks must provide fidelity (transmit the information without loss of changes) and reliability (less than 3 minutes down per year).

Nowadays, most of transmission is digital. A series of pulses is transmitted through a channel with attenuation, dispersion, interference from other signals and noise. Therefore what reaches the other side is considerably different and has to be estimated what the original input was.



As having a dedicated physical infrastructure for each service would be far to expensive and messy, the big majority of services share the same channel, the optic fiber. It is easily shared because of the available bandwidth

in it. Thus, the signals are multiplexed at the entrance, using different wavelengths (**Wavelength-Division Multiplexing (WDM)**) and de-multiplexed at the other end, to follow each one to their device that is requiring the service.

Note that WDM is exactly like **Frequency Division Multiplexing (FDM)** but in the optical domain. Technically they are exactly the same as changing the wavelength is nothing more than changing the transmit frequency.

A single mode optical fiber can reach throughputs of 10 Terabits per second.

Remember that there are many ways of scheduling frames in a multiplexer. With time slots or doing it statistically are two ways. Also, there are 2 types of switching: packet switching and circuit switching. Circuit is when a channel is constantly reserved for a certain application even if it is not being used. Packet switching allows a much better share of the resources. Packet switching principle is based on sending packets whenever there's a packet to transmit and use all the resources to do so as fast as possible. Therefore, the "speed" of the internet depends a lot on the amount of people that are accessing it.

Regarding the physical infrastructures for the transmission of data, those go from satellites, well the open space in general as microwave links are also a thing, twisted pairs, optical fibers and even a few more that are less common.

Finally, a look at the tendencies is pertinent. The traffic is increasing constantly, at a rate of 30% a year, therefore the network must be upgraded as the time passes as well, or else it won't be able to handle the traffic of the future. Not only are the links being upgraded since now we have fiber to the home, terminating really in our router, but each node must be upgraded as well to cope with the traffic resulting in new switches, larger datacentres, ect... However, all of this must be standardised to guarantee compatibility between countries, operators, manufacturers and users and to ensure minimum quality of service for all users. This standardisation is done by the International Telecommunication Union (ITU) that has two main sectors of interest: the -T sector regarding telecommunications in general and the -R sector for radiocommunications that is more focused in point-to-point, mobile, satellite links, ect... Additionally, ETSI, ISO, OSI, ANSI, IEEE are some of the main organisations that standardise technologies. IEEE is the best :)

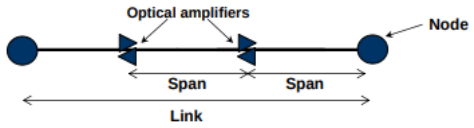
2.2 Networks Fundamentals

A network is composed of nodes and links and can be represented by graphs. However, a clear distinction between networks and graphs has been made in class: a network is a graph with a few more numbers that represent various network parameters. These parameters will be talked later, but can be delay of a link, distance, ...

The physical topology concerns the physical connections that are in place while the logical topology for a certain case concerns the actual flow of information. Even though every computer is connected in a network, maybe the information always flows from one to the others and the graph that represents that has much less links.

A link can be unidirectional or bidirectional. If the link is unidirectional, sometimes is referred to as an arc. $e_1 = (v_1, v_2)$ is the representation of a link, and the order of the nodes matter if it's an arc.

In optical fiber networks, or other networks that require amplifiers, the space between amplifiers (distance the signal has to go attenuating) is called a span.



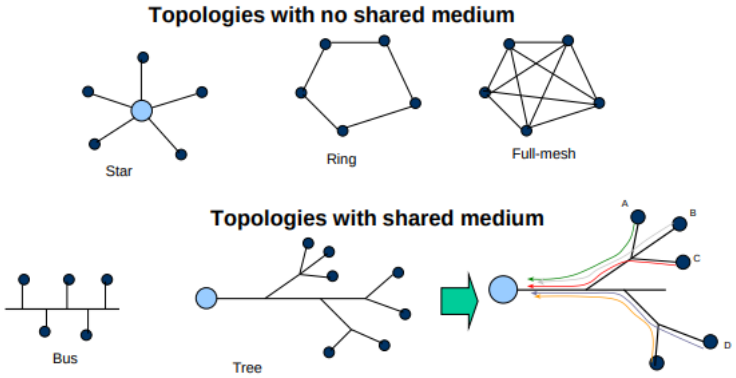
In a graph there's N number of Vertices(v_i), and L number of Edges(e_j). And the degree of the vertex is the number of edges it has. It's called the order of the graph, it's number of vertices, and the size of the graph it's number of edges.

Directed graphs only have unidirectional links, while undirected graphs only have bidirectional links.

The reason to make the distinction between directed and undirected graphs (unidirectional and bidirectional edges): in case of optical fiber, which is what connects most of long distance networking, is required to use more than one fiber. Because and optical emitter can't receive as well (at least in the same fiber). Also, if amplifiers are required, note that they are directed as well.

A path can be represented by a set of links, starting at some node. Source and sink are the names for the first and last vertex of that path.

2.2.1 Network Topologies



Bus, Ring and Star are the main physical topologies. Tree as well.

A tree is simply a graph with no cycles.

2.2.2 Network representative Matrices

A graph can be represented with an **Adjacency matrix (A)** , with $a_{ij} = 1$ if there's a direction from the vertices i to j.

The average node degree is given by the average of each node's degree which won't be more than the sum of all links, times 2 divided by the number of nodes. Times 2 because each link contributes for the degree twice, once at each end.

$$\delta_i = \sum_{j=1}^{N} a_{ij}$$

The average node degree is given by

$$<\delta> = \frac{1}{N}\sum_{i=1}^{N}\delta_i = \frac{2L}{N}$$

The Network diameter (D_R) is the maximum number of links between nodes through the shortest path between them. D_R is the longest of the shortest paths between every node.

Every link can have an associated cost (a function of distance, delay, reliability, actual cost, or other parameters) and a capacity (u_e denotes the capacity of node e).

The link capacity can be measured in any traffic unit that is appropriate for the problem. In packet networks: bit/s; in SDH networks: STM-N; in OTN networks: OTU-k; in WDM networks: number of wavelengths

Despite similar to an Adjacency Matrix, the **Demand matrix (D)** is slightly different. $d_{ij} = 1$ if the traffic flows from the vertex i to the vertex j.

Note that the diagonal of this matrix should be empty, or else it would mean that a certain node would receive information from himself, which makes no sense.

The mean number of demands is the number of demands divided by the number of nodes.

$$<d> = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}$$

The number of unidirectional demands (edges) in a case of full mesh logical topology is $D_1 = N(N-1)$. N nodes x the other N-1 nodes. Note that One other way of seeing it is that the D matrix is $N \times N$ but we need to take N away due to the empty diagonal. $D_2 = \frac{D1}{2}$ is the number of bidirectional demands, which is when only the top triangle of the D matrix is considered. This is usual because with bidirectional links the D matrix will always be symmetric.

Another interesting matrix is the **Traffic matrix (T)** and it's used to denote traffic intensities. It only has entries different from 0 in the exact same places the Demand matrix has. It's used for static traffic designs.

In transport networks the traffic units is the type of client signals: Ex: E3 (34 Mb/s), STM-1 (155.51 Mb/s), GbE (1 Gb/s), 10 GbE (10 Gb/s) ,etc. This traffic units must be converted to traffic units appropriate to be used in network design. These traffic units must be VC-n in SDH networks and in OTN networks ODU-k signals.

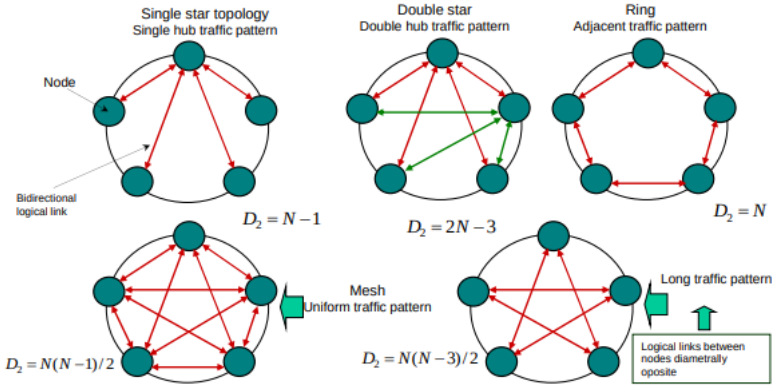
Considering now a Dynamic Traffic case, a few concepts arise:

- Average intensity of data flow between two nodes;
- Traffic bursts are time intervals where the flux of data is considerably higher than the average rate;
- Peak rate is the maximum instantaneous intensity.

Aggregation level or multiplexing level reduces the traffic *burstiness* because there less often flows get fully used and is more likely that the information can flow at a constant pace instead of in bursts.

Note one important distinction between Logical and Physical Topologies that we haven't done yet is that there can be many logical topologies on one physical topology. This can happen in the following way:

- **In a ring network there are diferent ways how the traffic can flow in a network leading to different network topologies (single star, double star, ring, mesh, etc.). *N*: number of nodes; D_2: number of bidirectional (two way) logical links.**



Routing is how a packet travels in a network. Therefore, routing ends up being the map between logical and physical topologies.

In optical networks the path between two nodes is usually called "lightpath".

We can also define a **Cost Matrix (C)** where each element c_{ij} represents the costs between nodes i and j.

The path can be performed manually (static routing - Demand matrix is time invariable) or dynamically, through routing algorithms (dynamic routing - Traffic matrix is time dependent, with constant arrival and termination of new demands).

Additionally, if the a given traffic demand(connection) is able to use more than one route, it is called a multipath routing process, else it is a mono-path process. Because there are usually many paths connecting two nodes, some metrics are taken into account when choosing which to follow:

- 1) Minimizing the network cost;
- 2) Minimizing the traffic in the most loaded link;
- 3) Minimizing the number of hops (number of links in the path);
- 4) Minimizing the path distance;
- 5) Maximizing the protection capacity, etc.

Most of the routing strategies incorporate some sort of shortest path algorithm to determine which path minimizes a particular metric, which can be for example 1), or 3), or 4). Note that the same algorithm used to 4) can be applied to 3) making all link distances identical.

From the physical topology, described by a graph G(V,E) and the traffic matrix T, describing all the traffic demands to be routed, one can perform shortest path algorithms such as Dijkstra's algorithm.

Order the demands according to a certain sorting strategy:

- Shortest-first: The demands with the lowest number of nodes in its path come first in the list;
- Longest- first: The demands with the highest number of nodes in its path come first in the list;
- Largest-first: The demands with the highest number of traffic units come first in the list;
- Random ordering: the demands are not known initially.

Route de demands according to the orderings. To break a tie choose the path that minimizes the load in the most loaded link.

Dijkstra's Algorithm

Consider a generic node *i* in a network with *N* nodes from where one wants to determine the shortest path to all the other nodes in the network. Lets l_{ij} be length (cost) of the link between node *i* and node *j* and d_{ij} the length of the shortest path between node *i* and *j*.

Algorithm:

1) Start with the source node i in the permanent list of nodes, i.e. $S=\{i\}$; all other nodes are put in the tentative list labeled S'. Set $d_{ii}=0$ and $d_{ij}=\infty \quad \forall j \neq i$.
2) For all the adjacent nodes to i set $d_{ij} \leftarrow l_{ij} \quad \forall j$ adjacent to i .
3) Identify the adjacent node j (not in the current list S) with the minimum value of d_{ij} (permanent node), add it to the list S ($S=S\cup\{j\}$) and remove it from ($S'=S'\setminus\{j\}$). If S' is empty stop.
4) Consider the list of neighboring nodes of the intermediate node j (but not consider nodes already in S) to check for improvement in the minimum distance path by setting $d_{ik} \leftarrow \min(d_{ik}, d_{ij}+l_{ij})$. Go to Step 3.

Now is pertinent to introduce yet another matrix, the **Hop Matrix (H)** where each element h_{ij} denotes the minimum number of hops from node i to node j.

The average number of hops per demand is nothing more than the sum of the hops of all demands divided by the number of demands. The number of demands was set as the unidirectional links between two nodes. Therefore, if 2 nodes share information between themselves (don't need to have a physical link, a logical one is enough) then there's a demand.

Therefore, the average number of hops can be computed:

$$< h >= \frac{1}{D}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} h_{ij}$$

Note that coherence is key here. If the amount of demands are the bidirectional demands, then the number of hops considered should only be the top half of the hop matrix. **If links are bidirectional, then there will always be a symmetry in these matrices** and we should compute the average with amount that mean the same thing.

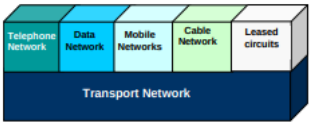
As means of simplifying this calculation, because hops and demands can be dynamic, a way of having a notion on the order of magnitude and get a fairly good approximation, when the number of nodes N is $4 \leq N \leq 100$ and the average node degree $< \delta >$ is $2.5 \leq < \delta > \leq 5$, is by computing the semi-empirical relation:

$$< h > \cong 1.12\sqrt{\frac{N}{< \delta >}}$$

2.2.3 Layers

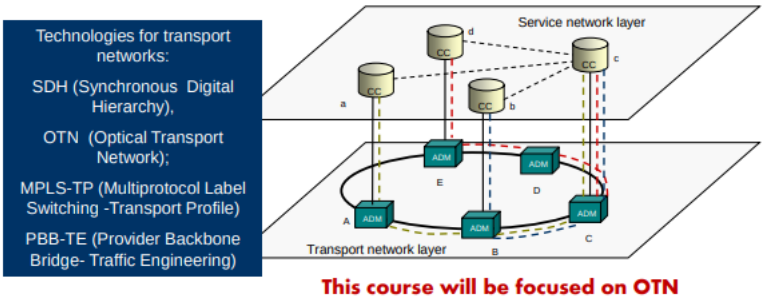
Typically, there's a layered structure in the network. The layer above acts as a client of the layer beneath and each layer appears a black box that supplies a service to the layer above.

The service layer is the one closer to us.



Add/Drop Multiplexing(ADM) are multiplexers controlled by **Control Centres (CC)** that decide what to add and what to drop form the fibre. Note that these don't manage the network, they just use it.

Nowadays, apart from local networks, everything is connected with fibre. Therefore, is pertinent to mention 4 key technologies for transport networks:



This course will be focused on OTN

SDH is also used in Hertzian links, MPLS is Multiprotocol Label Switching and is very useful for routing through different technologies and to choose more carefully the paths. As said above, OTN will be our transport technology.

OTN has become the new standard for a while now and has a few differences compared with SDH. The most important of which is distinction between fixed frame size and fixed rates. SDH has a fixed rate while OTN can increase it's rate to match the client's and this is very important for scalability and being more future proof

A very important distinction between the transport and the service layers is that the representation in the service layer has nodes connected with **logical topologies** while the transport layer has nodes connected with **physical topologies** .

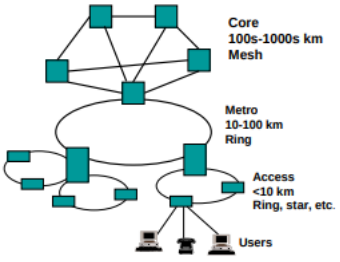
OTN	SONET/SDH
Asynchronous mapping of payloads	Synchronous mapping of payloads
Timing distribution NOT required	Requires right timing distribution across networks
Designed to operate on multiple wavelengths (DWDM)	Designed to operate on multiple wavelengths
Scales to 100Gb/s (and beyond)	Scales to a maximum of 40Gb/s
Performs single-stage multiplexing	Performs multi-stage multiplexing
Uses a fixed frame size and increases frame rate to match client rates	Uses a fixed frame rate for a given line rate and increases frame size (or uses concatenation of multiple frames) as client size increases
FEC sized for error correction to correct 16 blocks per frame	Not applicable (no standardized FEC)

The network management systems sends configurations through the **Data Communication Channel (DCC)** . Moreover, not all parts of the network are the same!

In the core and metro networks the physical topology is normally determined by the transport layer.

The access network uses a large variety of technologies and topologies and is responsible for the largest fraction of the operator´s investment on a telecom network.

Transmission technologies in an access network: twisted pair (copper), coaxial cable, optical fibre, wireless solutions (FWA).



One important distinction in terms of how things are connected physically is evident comparing the access networks with the other parts of the network. Access Networks, as opposed to what happens to the rest of the network, uses twisted pair or one optic fiber. This is because of the cost versus the bandwidth a fiber offers.

One fiber can carry several times the traffic of one user, therefore can be used for multiple users. Moreover, very often the final leg is done with twisted pair due to cost reasons.

In a more abstracted way, one can identify three planes:

- **Data Plane : is concerned with the transmission of the data between the users (forwards the traffic). Assures the physical support.**
  Also called forward or switching plane
- **Control Plane: is responsible for exchanging control information (signalling) between networks elements (nodes), which is used to set up, maintain, and tear-down connections. Examples of control planes: Signalling system nº 7, GMPLS (*Generalized multiprotocol label switching*), SDN (Software-Defined Networks) etc.**
- **Management Plane : Consists of several functions like detecting (alarms) and repairing failures (fault management), network element configuration (configuration management), performance monitoring to ensure to clients quality-of-service (performance management), allowing the system administrator to control personal access (security management).**

As you already know, there are circuit switched or packet switched networks.

Circuit Switched require circuit establishment and tear-down at the beginning and at the end, respectively. However, a distinction is made between physically *switchable* circuits and semi-permanent circuits. The first ones, a physical circuit is easily switched to connect one end to the other, while the second type regards circuits that are more static, that are much more difficult to switch. The semi-permanent must be switched by the administrators in order to attribute these circuits to a user for a somewhat long period of time (not just one transmission).

Circuits can be switched or semi-permanent. The first ones are established by the control plane (by signalling) as the case of phone circuits. The second ones are established by the management plane as it is the case of the electrical paths in SDH networks, or optical channels in OTN networks.

The biggest difference is the amount of time each one uses de circuit for. Switched circuits is around minutes, Semi-permanent are months.

Then, of course, there are packet switched networks that allow a much better share and efficient use of resources.

The key notion to have is that all services use the transport network! This is the highway of data! Nowadays it is impossible to have dedicated physical connections for each service. And this section of *Networks - Transport Networks* is based on that!

Telephone Networks

Telephone Networks before used circuit-switching. Note that the only ones that use circuit-switching are the landlines! Nowadays, our mobile communications are able to replace this fixed circuits lines

Local exchanges (Access) - are small switching centres that serve a small area.

Transit exchanges(metro, core) occur in **Primary trunk exchanges** is used to transfer the traffic and to interconnect several circuits.

They usually have a physical connection between them, but not one of their own. It wouldn't be viable to have dedicated circuits. It's here that the transport networks comes.

Different components have different topologies. The network is a hierarchy, having different topologies in different parts of the network.

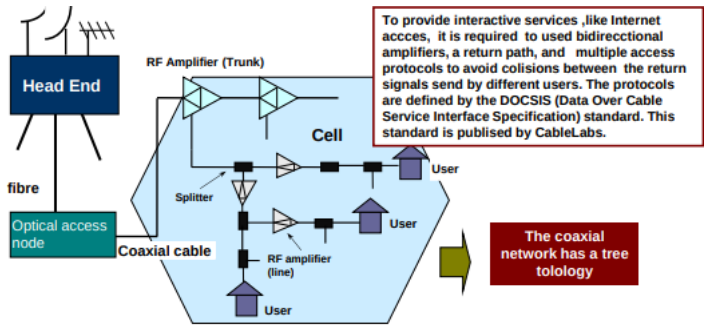
Digital Circuit-switching meaning that the data being switched is of digital nature.

Hybrid Fibre-Cable Network

There's an head end that distributes all the channels for all the users.

The network is called hybrid because it uses both fibre and coaxial cable.

Excellent example of a tree network. Like all the networks with share mediums, it required multiple access protocols to allow everyone to use the medium (FDMA and TDMA are 2 of the multiple access technologies).



The transport network is represented by the optical fibre that connects the head head to the optical access node. Like the cellular communications, one optical access node covers a certain area, with coaxial cable.

IP Networks

There are 2 ways of sending IP packets:

- Without Connection, it is necessary to have a buffer to reassemble all the IP packets. The packets are simply send to the network in a best effort way.

- With Connection: when certain QoS is required, then MPLS is used, so that certain resources can be reserved and the QoS met. MPLS establishes a virtual circuit before starting transmitting the packets. MPLS acts between layer 2 and layer 3. It places a label between the layer 2 and layer 3 labels.

There are 2 types of MPLS routers: - LER - Label Edge Router are the ones that put the labels and take them out; - LSR - Label Switching Router simply take labels out.

MPLS works by pre-establishing a path for information flow. Then, at each hop, according to the label id's available in each LSR the label arrives with one number and it leaves with other, based on the MPLS routing table in each MPLS capable router. This table entries are created based on the MPLS protocol and pre-path find procedures.

This is how a MPLS table looks like:

	Input port	Input label	Ouput port	Output label
FEC 1	1	15	2	10
FEC 2	2	25	3	20

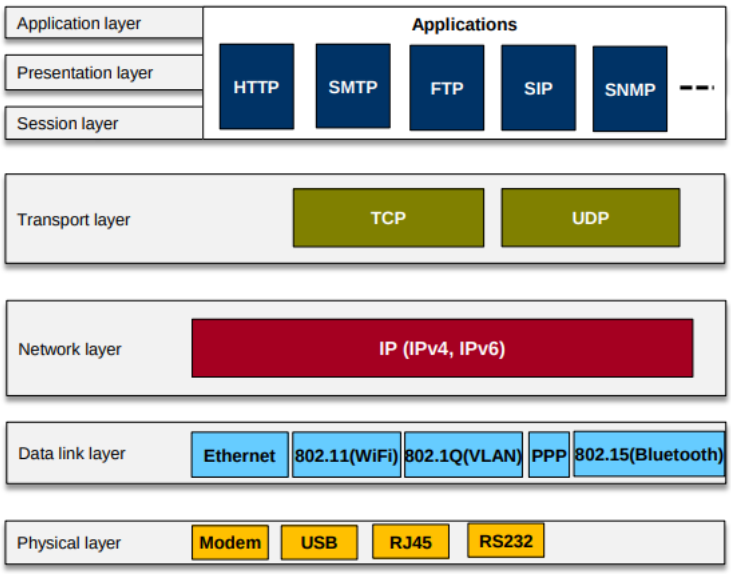
Generalized MPLS (GMPLS) is the *de facto* (practice that exists practically even though it is not formalized by law) of the control plane of the Wavelength Switched Optical Network (WSON).

Label Switching allows Traffic Engineering:

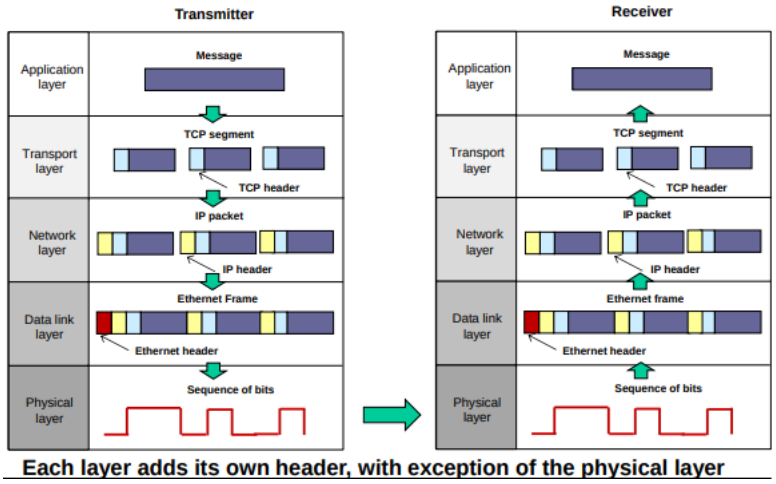
Traffic Engineering (TE) deals with a set of procedures required to optimize the performance of telecommunications networks and use network resources in an efficient way. It permits, for example, to route the traffic in order to avoid congestion and to maximize the transported traffic.

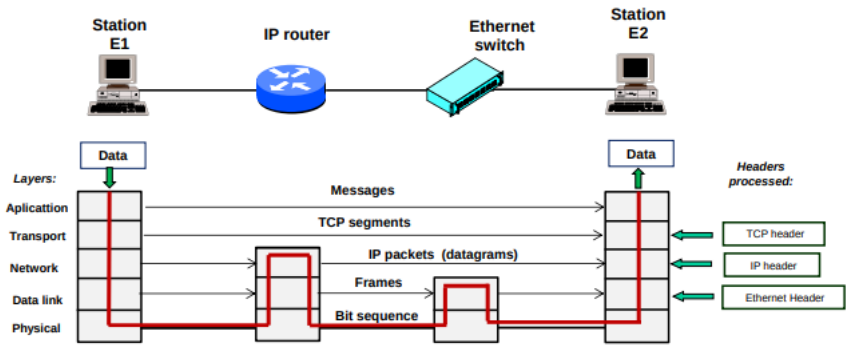
2.2.4 Layered Model Overview

Open Systems Interconnection Model:



Transport layer connects applications. Network layer connects machines with different IPs. Data link layer connects two machines with their interfaces' MAC addresses. Physical layer is what is responsible for putting the data into a cable or into the air to perform the actual transmission. For instances, spectrum, modulation and coding techniques and intervals of transmission.





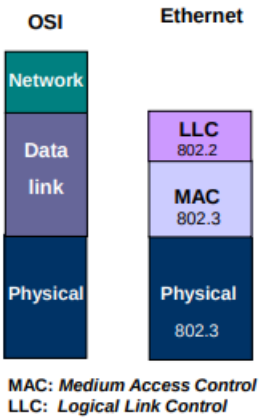
2.3 Ethernet Networks

Most of today's network traffic is generated from Ethernet interfaces. Made from twisted copper pair, through slightly more advance techniques and protocols is still possible to get quite a good throughput with a low cost cable.

Uses CSMA/CD Carrier-sense multiple access with collision detection because it can sense if there's anyone using the bus at all times. WiFi for instances CSMA/CA is used since we may be causing a collision we can't detect.

The Ethernet protocol 802.3 is used in all parts of the transport layer, not only at the access level and specially not only in LAN (Local Area Networks). Note that the Ethernet protocol was standardised by IEEE and the standard specifies not only the physical specifications but also Data Link Layer frame formats. WiFi 802.11 is another protocol that includes physical, like transmit powers and modulations, and Data Link specifications

A rule of thumb is to use optical cables for distances above 100 metres.

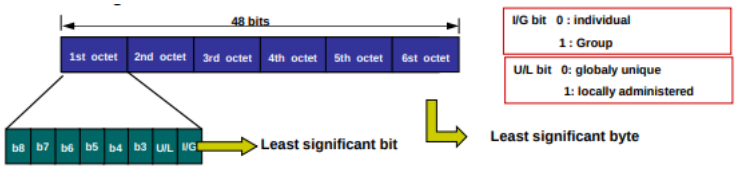


- The LLC (Logical Link Control) sub-layer is responsible for the flow and error control between the nodes.
- The MAC sub-layer is responsible for the media access control, addressing, error detection, frame delimitation, by organizing the bit sequences into frames.
- The physical layer deals with the bit transmission and reception, with the electrical, optical and mechanical properties of the interfaces, with the type of connectors used, etc.

The Data Link layer is responsible for frame processing, and error detection through the calculation of a CRC (Cyclic Redundancy Check). If a frame has errors, it is discarded. Most of error correction is done higher up in the stack. TCP guarantees error free transmissions when it says they were successful.

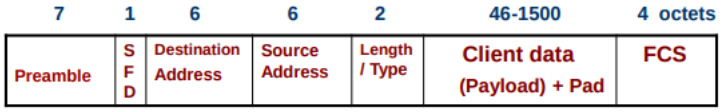
MAC stands for Media Access Control. Thus, the MAC addresses are addresses to access the physical media. CSMA protocols are placed in this layer.

There are 2 bits in the MAC address, the two least significant ones form the first octet.



U/L refers to the uniqueness of the interface. The MAC address can be changed becoming a local one. I/G refers to one interface only or a group that has that MAC address that received always the same thing.

Write *ifconfig* or *ip a* in linux (or *ipconfig* in windows) is possible to see this MAC address.



Preamble: sequence of 7 octets (0101....) permits the recovery of the signal clock in the receiver, when it operates in burst mode.
SFD (*Start of Frame Delimiter*): Pattern of 8 bits (10101011) that indicates the beginning of the frame.
The destination and source address are fields with 6 octets.
Length/type: sequence of 2 octets to indicate the length of the data field (≤ 1500) or the type of frame (≥1536) (ex: data frames (IPV4, IPV6, MPLS, etc) , 802.1Q, 802.1ad, control frame (faults, flow, etc.))
FCS (*Frame Check Sequence*): Uses a four-octet CRC code calculated over al the the octets apart from the preamble and SDF fields.

Because there are only 1500 bytes to address (to count) with the Length/Type field, then there will be quite a few empty bits in this field. The rest of the bits are used to denote the type of the frame:

If the value in the field lies from 0 to 1500 it indicates the length of the Date field . If the value ranges from 1536 to 65 535 it indicates the nature of the Data field. Examples of field types:

0x0800: IPv4; 0x8600: IPv6; 0x8808: Ethernet flow control; 0x8847: MPLS unicast; 0x8848: MPLS multicast; 0x8100: IEEE 802.1Q; 0x88A8: IEEE 88A8.

The purpose of the Preamble is to achieve clock synchrony.

2.3.1 Multiple Access

TDMA, FDMA (WDMA), CDMA all have the same principle: divide signals in a given domain.

TDMA divides signals in the time domain: different transmissions occupy different time slots.

FDMA divides in frequencies, WDMA divides in wavelengths. The only difference is that one is more used in wireless applications and the other with fibres.

OFDMA starts combine the previous ideas: divides in the time and in the frequency domains, therefore is able to attribute resource blocks with much better efficiency. A use of OFDMA is in 4G and 5G where a physical resource block is nothing more than a set of sub-carriers used for a given time period, usually called one TTI (Transmission Time Interval).

CDMA is used in 3G and with GPS, so that different satellite can transmit all at the same time.

Then there's another category of multiple access techniques like CSMA. Namely CD and CA, collision detection and collision avoidance.

Collision Detection is when a collision is possible to be detected and after being detected certain steps are taken to minimize the risk of happening again.

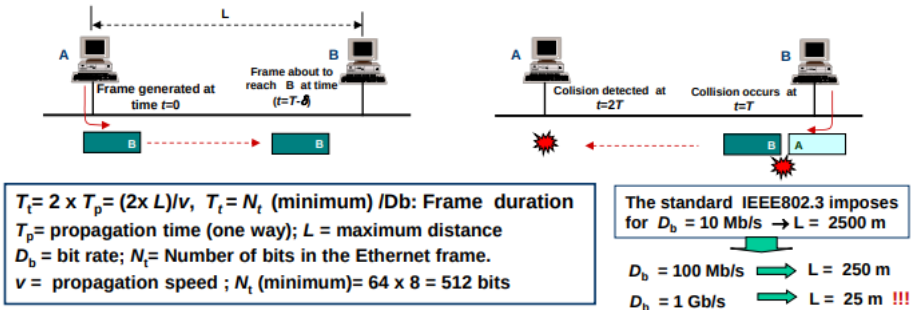
Collision Avoidance is when the collision can't be detected and should at all costs be avoided.

A curiosity: he speed of light inside an optical fibre, considering an average glass refractive index of 1.5, is:

$$v = \frac{c}{n} = 2 \times 10^8 m/s$$

In cabled Ethernet, CSMA/CD is used. Disregarding all other phenomena that limit the debit - dispersion that leads to distortion, attenuation, etc. . . - the use of CSMA/CD is limited by the distance that the information can travel before one discovers that a collisions as occurred. In essence, if the maximum debit is 10 Mbit/s and the maximum ethernet frame is around 1500 bytes long, it will take $\frac{1500*8bits}{10\times10^6 bits/s} = 0.0012s$.

The electrical signals propagate at speed of light - don't confuse with the speed of light in the fibre, in the fiber the light is propagating in glass, here there's a wave propagating in TEM mode along a twisted pair or coaxial cable. Therefore, in order to notice a collision, the signal must be able to go all the way to the other end and comeback BEFORE the sender is ready to send another frame. Mathematically: $T_{max-frame} \geq 2T_{propagation}$. This is because, if there is a collision at the other end, the sending machine must receive that collision before finishing the transmission, otherwise it would keep sending frames without knowing about the collision.



Since $T_{propagation}$ is proportional to the length of the connection, and the time of transmission is inversely proportional to the bit rate, then the bigger the rate, the smaller the maximum distance between machines can be.

That is also why CSMA/CD is not used when the rates get too high!

Type	Bit Rate	Mode	Topology	CSMA/CD	Medium
Ethernet	10 Mb/s	Half-duplex	Bus	Yes	Coaxial cable
Fast-Ethernet	100 Mb/s	Half e full duplex	Star	Yes	Copper + Fiber
Gigabit-Ethernet	1 Gb/s	Half e full duplex	Star	Yes	Copper+ Fibre
10 Gigabit Ethernet	10 Gb/s	Full duplex	Star	No	Copper+ Fibre
100 Gigabit Ethernet	100 Gbit/s	Full duplex	Star	No	Fibre

Half-duplex → CSMA/CD (Carrier Sense Multiple Access/Collision Detection)
Full-duplex → Switched Ethernet

Note that this is only necessary if a medium is shared! If it is dedicated, it is not necessary.

A switch is a layer 2 - the Link Layer - equipment that commutes Ethernet frames,

Switches with higher capacities are higher up in the grid, closer to the core, because they have to handle many more traffic flows.

Some functions of switches:

- Forwarding - Simply check the destination, it knows to which port that destination is connected to and send the frame in that port destination is
- Broadcasting - Sending to all destinations, this happens when it doesn't know where to send it and needs to find out.
- Filtering -

In order to avoid problems such as exponential spreading of frames, a **Spanning Tree Protocol** at a logical level is used. Closed loops go away with this protocol since a tree is created and all notes still are connected since it is a spanning tree. BPDU - Bridge Protocol Data Units are the frames sent to establishing this tree.

Ethernet frames don't have time to live. But in BPDU's have TTL

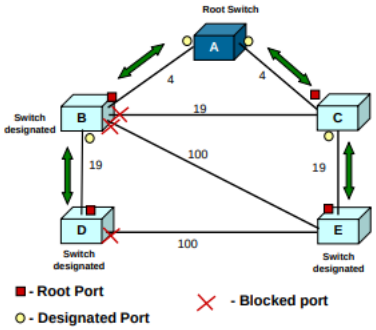
This Spanning Tree Protocol can be implemented in the following way:

1. Root Switch election - typically the one that has the smallest MAC in the network because computing the one that has the smallest distance to every other node is too troublesome. Therefore, after receiving a BPDU with an ID lower than the current root, it replaces his belief of root and propagates that information.
2. Convergence to Spanning tree starting at the Root - The root propagates BPDU's with a cost to root (if the root is sending, the cost is 0). The nodes that receive do the same: **(root id, my id, cost to root)** . - If the received packet tells a cost to root that is smaller than before, that port is designated as the root port.

   - If the received packet tells a cost that is bigger, than that port will be blocked: not packets should be sent that way!

   - The ports that are not blocked and that will connect the lower nodes to the root will be called **designated ports** . These are the ports that node can send things to.

This is an example of the status of each



It is provided a bidirectional path with a tree logical topology between the root and all the switches of the network without closed loops

STP switch port states:

State	BPDU Rx	BPDU Tx	Data Rx	Data Tx
Disabled	No	No	No	No
Blocking	Yes	No	No	No
Listening	Yes	Yes	No	No
Learning	Yes	Yes	Yes	No
Forwarding	Yes	Yes	Yes	Yes

The listening state corresponds to the spanning tree configuration and building.

In the blocking state the port continues to receive BPDUs with certain periodicity (default 2 s). No user data is sent or received over the port. When an active port fails it may be necessary to activate the blocking ports.

How is the switch able to build the Source Address Table (SAT)

It can be filled by hand, by the network administrator. Or it can learn the MAC that communication to each port.

How packets Travel in the network We write a message in the phone or in the laptop and press send in the application (layer). First, there is some compression and encryption done by the application layer right away. Then in the application code, there is a section that is defining TCP or UDP sockets - very likely TCP -

How to build minimum spanning trees? They are needed for the spanning tree algorithms like the Link Layer LAN minimum spanning tree algorithm.

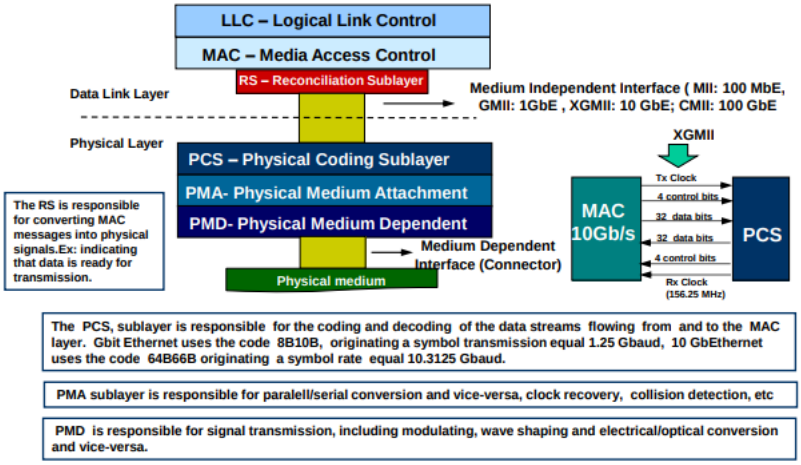
With the Kruskal's Algorithm:

1. order the edges by their cost, minimum first.
2. keep adding edges to a the spanning tree, as long as the edge doesn't create a loop (connects two nodes already connected in another way).
3. stop when the tree has **n - 1** edges.

2.3.2 Physical Layer of the Ethernet

The Ethernet protocol includes physical specifications.

Between the link layer and the physical layer (the layer right before the physical medium) there's a **medium independent interface (MII)** . In the picture below, depending on the debits being transmitted, this interface has several names. It has different names because it is a parallel interface that depends on the debit.



The first sublayer of the physical layer is coding. It will had signals to the bits. Each bit will

The second sublayer is responsible for the parallel/series conversion.

The last sublayer must generate the signal into the medium being optic fibre, twister pair, etc...

Code - mBnB

Firstly, the code. Required for robustness of transmission and to send symbols at lower frequencies than the actual bitrate.

m input, n output bits

By switching between Mode 1 and Mode 2, the number of 0's is identical to the number of 1's, which is important to keep the stochastic balance between 1's and 0's - important for detection.

3B4B Code

Input bits	Mode 1	Mode 2
000	0010	1101
001	0011	0011
010	0101	0101
011	0110	0110
100	1001	1001
101	1010	1010
110	1100	1100
111	1011	0100

After the coding, a 1Gbit/s bit rate is converted into a $\frac{n}{m} * Rate$. For 3B4B, it will be necessary to transmit 4 bits in the time it took to transmit 3. Before, the rate per bit would be $\frac{1Gbit/s}{3}$. And this needs to be multiplied by 4 to have the 4 bits going at the same time. Therefore, 1.(3) Gbaud (symbols per second) is the new rate.

The ethernet transmission is done in baseband, no modulation.

Format used in the interfaces: [Value1] Base [Value 2]

Value1 = Bit rate for the transmission : Ex: 100 ⟹ 100 Mb/s

Base = *Baseband Mode*: Base band transmission (no modulation)

Value 2: Type of cable (T: *Twisted Pair*, F, X, R: Optical Fibre)

Examples:

10 Mbit/s Ethernet

10BaseT: Uses UTP twisted pairs of category 3 ou 5; max distance = 100 m
10Base F: Uses multimodal optical fibre

100 Mbit/s Ethernet (PCS: 4B5B)

100BaseT: Uses UTP twisted pairs of category 5; max distance = 100 m
100Base FX: Uses multimode optical fibre (62.5 µm); max distance = 2000 m

Gigabit Ethernet (PCS: 8B10B)

1000BaseT: Uses UTP twisted pairs cat. 5e; max distance = 100 m
1000Base SX: Uses multimode optical fibre (62.5 µm); max distance = 275 m
1000Base Fx; Uses single mode optical fibre; max distance = 5000 m

For the correct names Ethernet phy layer .

Avoiding Crosstalk

Electromagnetic interference becomes a limiting problem in very high frequencies. To reduce these interferences, a metal sheet foil can be used. A more effective solution is really to use a shield Therefore, shielding is necessary for twisted pairs.

The necessary bandwidth to transmit 10Gbaud/s, CHEEEEEEEEEEEEEEEECK HOOOOWWWWWW

Category 7 (CAT7) : Bandwidth equal to 600 MHz((@ 100m). Supports bit rates up to 10Gbit/s (10 GbE), using a cable with 4 STP pairs.

Gigabit + Ethernet

When the speeds are too big, either the distances need to be smaller or fibres must be used. Single mode even...

6 Python

6.1 Important Concepts

Python is a language with many characteristics, ones more obvious and easy to understand, like in-fix notation, others are more complex, like Dynamic Typing and Automatic Memory Allocation.

Moreover, many programming principles like anonymous functions, map, zip are important and should be addressed in order to achieve flexible and overall good programming skills.

Beforehand, check the Python Notebook and the slides from Rodrigo Ventura in the "Additional Material" Folder. They are very complete and give a proper insight on how everything works.

Tuples are faster than lists. So, if the idea is doing a very big iteration where there's no memory problem, converting to tuple before probably helps.

However, regarding speed, always use built-in functions and libraries! Almost always what is slowing the program down is a huge for loop... Libraries are compiled already and are very optimised, therefore by using them you are probably running very optimised C code, which couldn't be faster than it is.

6.1.1 Lambda and Anonymous functions

Lambda is the keyword used to make an anonymous function. The following 2 pieces of code do exactly the same.

```
    def my_key(x):
    return x[0]

    l.sort(key=my_key)


    OR

    l.sort(key = lambda x: x[1])
```

And a short example is:

```
    -----short example on how cool python is-----
    def make_multiplier(factor):
    return lambda x: factor*x

    f = make_multiplier(2)
    -----------------------------------------------
```

Therefore, lambda is nothing more than defining a function, without giving it a name. It helps keeping the code simple specially when the function is just going to be used once.

6.1.2 __main__

This can be perfectly understood in the following stackoverflow post .

In a nutshell, when a script is executed python assigns many names to certain variables like __main__. Is the script is executed in the terminal, then main will be the name of that script (without the .py). The *if __name__ == '__main__':* aims to distinguish between situation where the script was imported, so that the functions it contains can be used elsewhere (in this case the main name won't be the name of the script which is stored in __name__) and the situation where the script is directly called in the terminal. If the script is directly called, for it to do something, something has to execute and that's usually what goes inside that if. When the script is imported, most of the times only the definitions matter and the calls will be made in the script that is calling that one.

Therefore, is a good tool in case you want to make a script importable but also callable.

6.2 Some useful tools

To check if a variable points to a certain data type:

```
    isinstance(var, [list, tuple, int])
```

To define functions with optional arguments and call them in the incorrect order:

```
1
2       def draw_point(x, y, color='red', thickness=2):
3           print('x =', x, 'y =', y, 'color=', color, 'thickness=', thickness)
4
5       x = 1
6       y = 2
7       draw_point(x,y,'blue', 5)
8
9       draw_point(x,y,thickness=2, color='blue')
```

6.2.1 Unpacking Argument Lists

Unpacking Argument Lists

A very useful trick to pass many arguments at once.

6.3 Anaconda

Basically, it is the Python distribution. Because Python is so big, with so many packages and Python development is becoming quite big, a program to install things was created.

Go to the official Anaconda website and download the installation script for Python 3 and for Linux: anaconda.com/distribution

Install Linuxize - How to Install Anaconda on Ubuntu 18.04

6.3.1 Package Manager

User's Guide for Package Manager

Has instructions on how to install non-conda packages and many other useful things. It allowed the installation of streamlit!

6.3.2 Broken Jupyter

Tornado 6.0 breaks jupyter notebooks. Is required to uninstall it through pip and/or pip3 and through anaconda! Downgrade to 4.5.3. (This subsubsection will probably be removed in the future when it's fixed.)

6.3.3 Other

(base) Problem Ask Ubundu (base) in terminal - then close and open the terminal to take effect.

6.4 Pandas

Check this link :)

6.5 Jupyter Notebooks

Along with Anaconda comes a full installation of the most recent python version and the Jupyter Notebooks, which are awesome to write python.

Here are some very useful shortcuts to tame that beast:

There are 2 modes of shortcutting:

- The Command Mode (when border of the cell is blue). Press ESC to access this mode.
- The Edit Mode (when border of the cell is green). Press ENTER to access this mode.

The Edit Mode is clearly superior:

- Ctrl + Enter to run cell
- Shift + Enter to run cell and get directly to the next
- Ctrl + D to delete a whole line
- Ctrl + arrows jumps words (usual)
- Ctrl + Backspace deletes whole word (usual)

The Command Mode can be usefull sometimes, especially when adding cells is needed, but remember that the cell has to have a blue border:

- A - insert cell before
- B - insert cell after
- DD - delete current cell
- Z - undo cell deletion
- M - markdown input type (Check: Netbook MarkDown)

6.6 Spyder

Jupyter Notebooks is very fun for prototyping, however, to hard debug and develop some complex stuff, Spyder is the tool.

Install Anaconda. It is the absolute best way of getting everything up to date and running smoothly with the least complications. Check here for the complete tutorial on how to do so (very easy).

After installing Anaconda, check the Official Spyder Releases github page and install the most recent one. **Also remember never to run pip install spyder or similar or it can completely break Anaconda installation** .

Some useful shortcuts in Spyder:

- Add # % % to separate cells.
- Ctrl + Enter - Run cell
- Alt + Ctrl + Enter - Debug cell
- Ctrl + Shift + I - Go to Console
- Ctrl + Shift + E - Go to Editor

Note: spyder debugger is fucking shit. You can't get comfortably inside functions. VS Code does the job fairly well for the time being.

6.7 Keras - A powerful API for TensorFlow

Keras is simply the way of doing neural networks. It can't get easier than that! TensorFlow was created by Google and PyTorch, it's rival, was created by Facebook. Not only is Google a better company in terms of the use of NN to do things, but this website also agrees that TensorFlow 2.0 is a complete game changer since it also declares keras as it's main API, therefore making it very very easy to do complex computations.

Overall, two things are fundamental to have success with these tools:

1. To understand the functions and what all options mean, is necessary to have many mathematical expression in the head
2. Have a very good control of the API - Keras. This section will take care of this step, trying, as best as possible, to explain the first step as well.

However, learning how to use the api can take a while. Following are the main parts of a NN.

6.7.1 Basic Flow - Image Classification Example

Imports, data load, normalization, on mean and on stddev(not included in the example), create model, add layers, check summary, configure callback(s) and the optimizer, compile the model to apply the loss function and the optimizer, fit the model, predict with the model, plot training and validation loss curves, plot confusion matrices and accuracy scores opposing the predicted data with the actual test data outcomes.

```
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import tensorflow.keras as keras
from keras import utils, layers, models, callbacks, optimizers
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from sklearn.metrics import confusion_matrix, accuracy_score

x = np.load("mnist_train_data.npy")
y = np.load("mnist_train_labels.npy")

x_test = np.load("mnist_test_data.npy")
y_test = np.load("mnist_test_labels.npy")

print(x.shape)
print(y.shape)

plt.imshow(np.squeeze(x[5]))

#normalizing
x_norm = x / 255
x_test_norm = x_test / 255

#y to categorical
labels = keras.utils.to_categorical(y, num_classes=None)

model = keras.models.Sequential()

model.add(keras.layers.Flatten(input_shape = (28,28,1)))
#in case the samples are one dimensional, Input layer is enough.

#after the first layer it is not necessary to specify the input size
model.add(Dense(64, activation='relu'))
model.add(Dense(128, activation='relu'))
model.add(Dense(10, activation='softmax'))

#Expected: Layer 1 weights: (784+1) * 64 = 50240
#         Layer 2 weights: (64 +1) * 128 = 8320
#         Layer 3 weights: (128+1) * 10 = 1290
#                            Total  = 59850
model.summary()

"""
Example of CNN:
model2.add(Conv2D(16,kernel_size=(3,3), activation='relu'))
model2.add(MaxPooling2D(pool_size=(2,2)))

model2.add(Conv2D(32,kernel_size=(3,3), activation='relu'))
model2.add(MaxPooling2D(pool_size=(2,2)))

model2.add(Flatten(input_shape = (5,5,32)))
model2.add(Dense(64, activation='relu'))
model2.add(Dense(10, activation='softmax'))
"""



stopper = callbacks.EarlyStopping(patience=15, restore_best_weights=True)
#restore_best_weights - restore weights from best epoch

# min_delta may me zero because the epochs after the best only count if there's improvement

Adam = optimizers.Adam(lr=0.01, clipnorm = 1)

model.compile(loss='categorical_crossentropy', optimizer=Adam)

hist1 = \
    model.fit(x = x_norm, y=labels, batch_size=300, epochs=400,\
        verbose = 0, callbacks = [stopper], validation_split=0.3)

y_pred_labels = model.predict(x_test_norm)

y_pred = np.argmax(y_pred_labels, axis=1)

plt.plot(hist1.history['loss'])
plt.plot(hist1.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')


```

```
sns.heatmap(matrix,annot=True,cbar=True)
plt.ylabel('True Label')
plt.xlabel('Predicted Label')
plt.title('Confusion Matrix')

print('Accuracy:', accuracy_score(y_test, y_pred, normalize=True)* 100, '%')
```

6.7.2 Sequential Model

There are Sequential and Functional Models. Functional is when the mess it too big. Very likely not be needed since Sequential can do much more that I know how to do, including image analysis, convolutional and recurrent NN. To learn the basic steps of it: Keras-Getting started with the Sequential Model And to learn the detailed methods of it and their functions: The Sequential Model - all steps

6.7.3 An optimizer

An optimizer is one of the two arguments required for compiling a Keras model - optimizes the casual gradient descent algorithm to achieve a faster convergence, i.e to find the minimum faster. We've seen a few methods on adaptive step size and momentums.

```
    model.add(Dense(32, input_dim=784))
    model.add(Dense(32, input_shape=(784,)))
```

They are equivalent and both simply mean: Add a fully connected layer with 32 outputs and 784 one dimensional inputs.

6.8 Plotting

Very similar to MATLAB, but different.

It all depends on all complex you want to go.

Way 1:

```
    plt.subplot(1,2,1) #n_rows, n_cols, index
    plt.plot(x,y) #or sns.heatmap...
    plt.xlabel("hey")
    plt.ylabel("no")
    plt.title("yello")
```

Way 2:

```
    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15,5))
    fig.suptitle('Confusion Matrices comparison')

    sns.heatmap(matrix,annot=True,cbar=False, ax = ax1) %#ax1.plot(x, y)
    ax2.set(xlabel='True Label', ylabel='Predicted Label')
    ax1.set_title('With EarlyStopping')

    sns.heatmap(matrix2,annot=True,cbar=False, ax = ax2)
    ax2.set_title('Without EarlyStopping')
    ax2.set(xlabel='True Label', ylabel='Predicted Label')
```

Way 3 - a great midterm between the other two:

```
    f = plt.figure(figsize=(10,3))

    ax1 = plt.subplot(1,2,1)
    sns.heatmap(matrix,annot=True,cbar=False, ax = ax1)

    ax2 = plt.subplot(1,2,2)
    sns.heatmap(matrix2,annot=True,cbar=False, ax = ax2)
    plt.title("helo")
```

In case is necessary to have plots of different sizes

6.9 Artificial Intelligence: A Modern Approach - Search Configuration

Clone the 3 repositories below.

Aima Code

Aima Data

ipythonblocks GitHub

Then put the ipythonblocks.py and the data folder inside the aima-python folder. That way the Jupyter notebooks should work

6.10 From Python 2 to Python 3

In Linux, if 2to3 is installed, one may simply run:

```
2to3 -w -n file.py
```

This will write the file translated into python 3 in the same file(-w) and without creating a backup(-n).

6.11 Good Practices for Python Code

This section is irrelevant if you only code alone for yourself. In case someone else will see your code, then you should write it as everyone else likes it. You'll probably agree with most if not all of these guidelines.

Style Guide for Python Code

Some of the ones I tend to violate the most:

- Variable names with underscore (underline)