Network Infrastructure

Simone Palumbo's Notes 2023/2024

MSc in Artificial Intelligence and Robotics

Introduction to Optical Networks

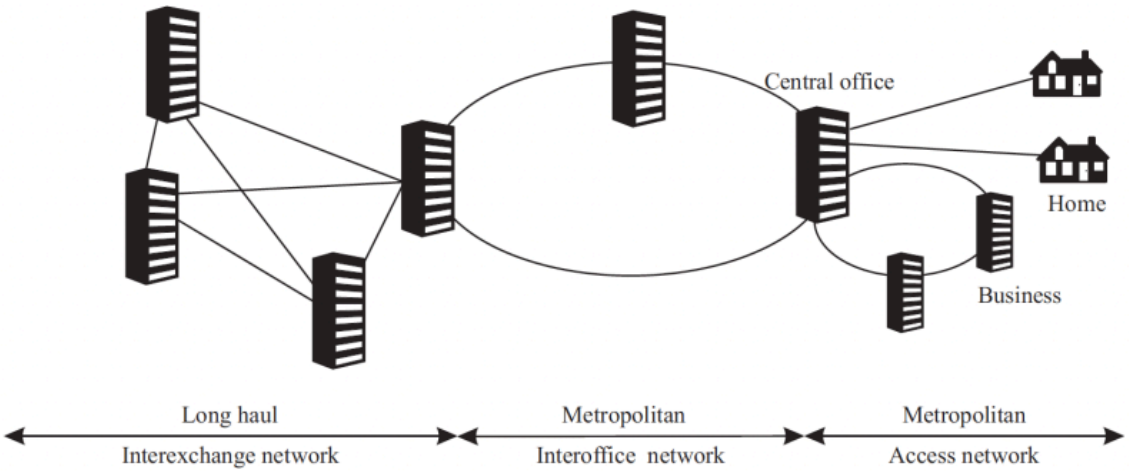
25/09, 29/09, 2/10

What is a Transport Network?

The goal of a Transport Network is to provide connectivity to clients that ask for a connection. TNs works with a connection oriented paradigm. The TN can be broken up into:

1. Metropolitan Network: the part that lies within a large city or a region
    a. Interoffice Network: connects central offices in the city / region
    b. Metro Access Network: from a central office (also called POPs[1]), it reaches individual businesses or homes
2. Long Haul, Interexchange Network: connects cities / regions.

Different parts of the TN may be owned by different carriers. Ring and mesh topology are used.

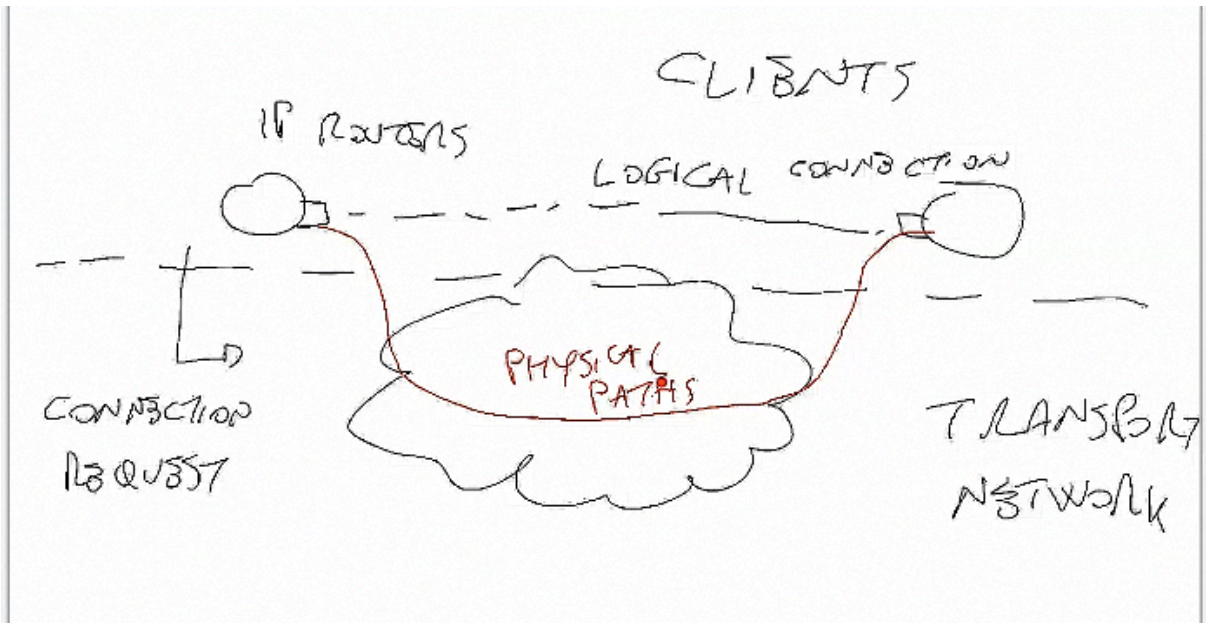


Client - Carriers interaction in the TN:

Let's see how two IP routers can be connected. We are talking about a logical connection between their NIC:

1. One of the two client performs a connection request to the TN
2. the TN create a virtual circuit over its physical network

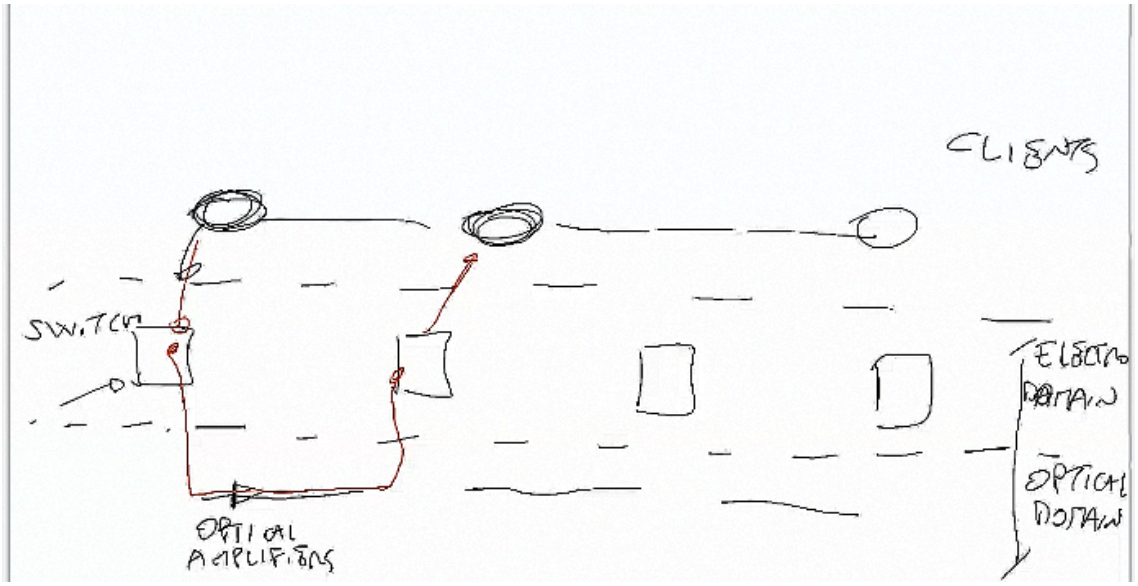
[1] point of presence



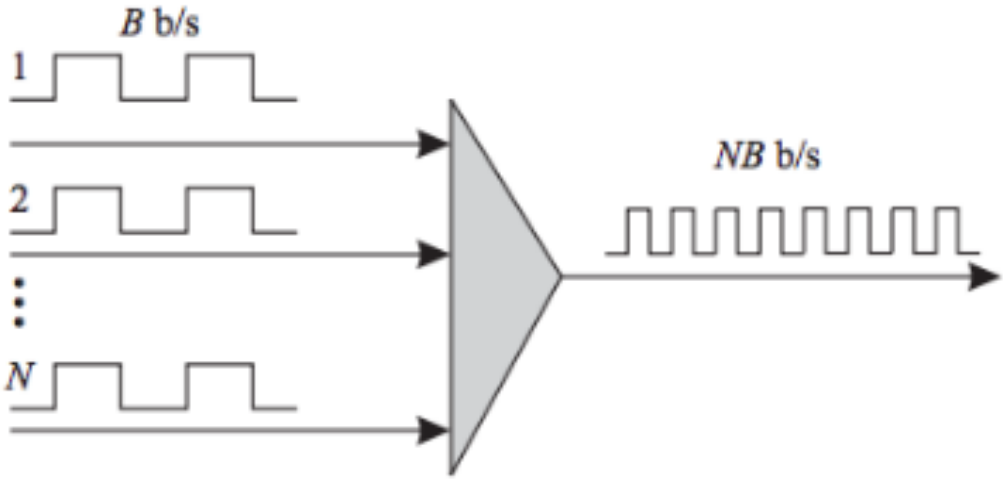
This way we don't have to build a physical cable to connect the two nodes. Moreover, we can share the physical infrastructure also for other connections.

Optical Technology to realize the TN and its Architecture:
A Transport Network is realized through optical technology. There are two main categories of Optical Networks:

1. First generation: optics used only for better transmission, possibly with optical amplifiers. Intelligent network functions, such as switching, are performed in the electronic domain, since to switch we use binary information and these are available only in the electronic domain.

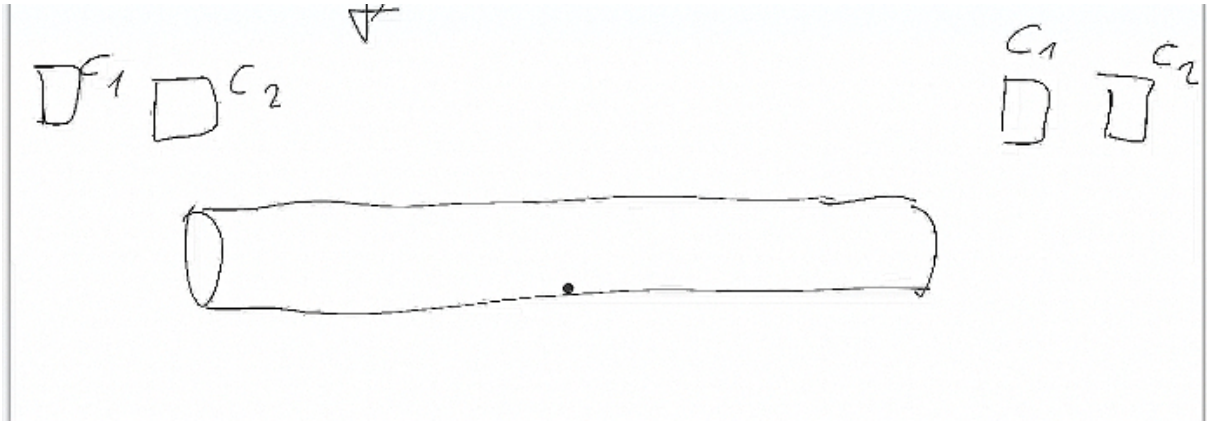


1st gen adopted Time Division Multiplexing (TDM), to assign statically the use of a slot to a specific user, so to always distinguish the users. This requires working in the electronic domain, since we need the concept of clock and bit rate.



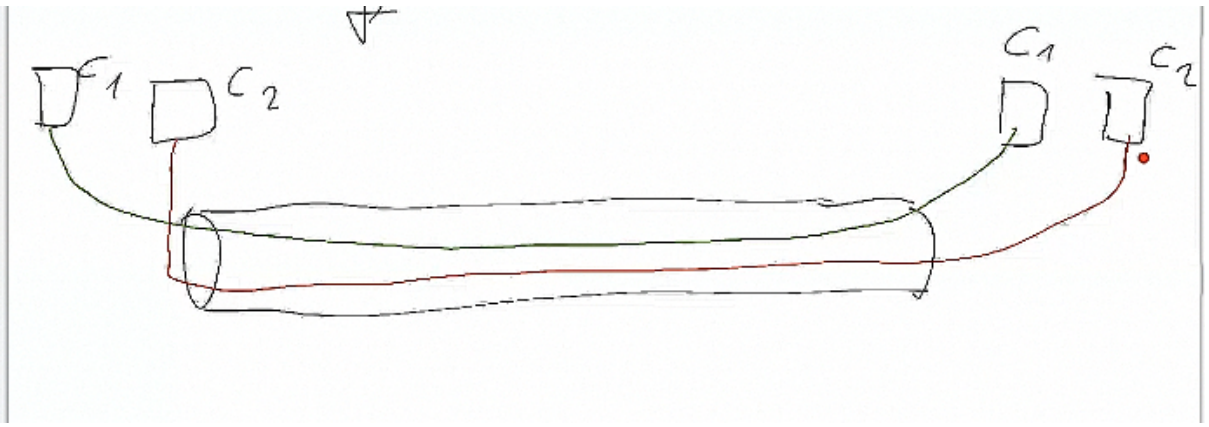
2. Second generation: also intelligent functions are done in the optical layer. We try to stay as much as possible in the optical layer for the performance benefits. The problem is that in the optical domain we don't have binary information and it's impossible to understand which is the source and which is the destination, since the TN is shared (e.g. if a light bulb it's transmitting we receive 1 if not 0, but we don't know which light bulb transmitted the signal).

   To solve this problem we have to find a way of switching a light ray. We use Wavelength Division Multiplexing (WDM), or FDM[2]:

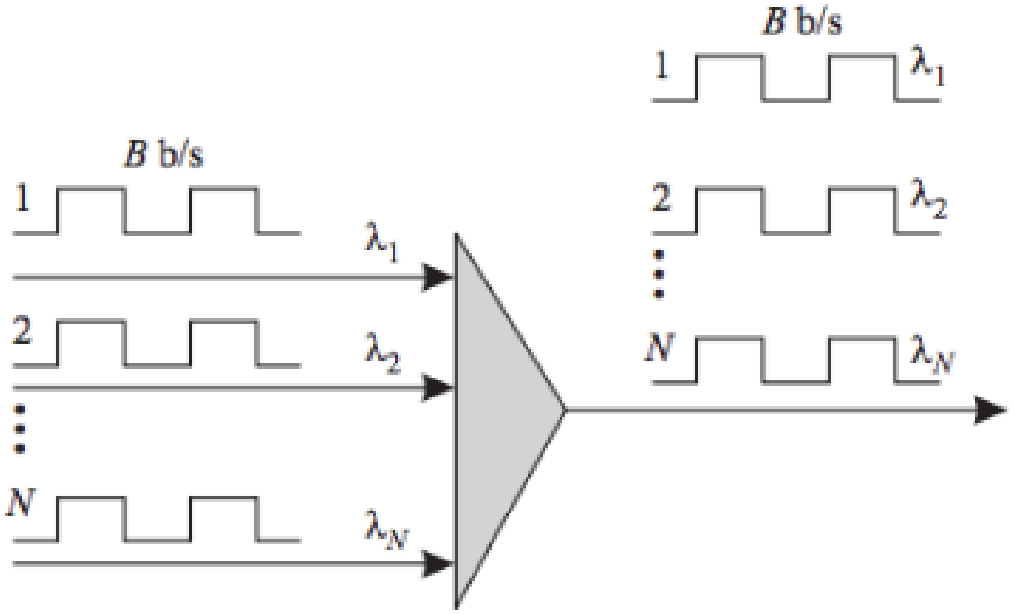


   since its light, we can think of frequency with color, so we assign c1 black and c2 red

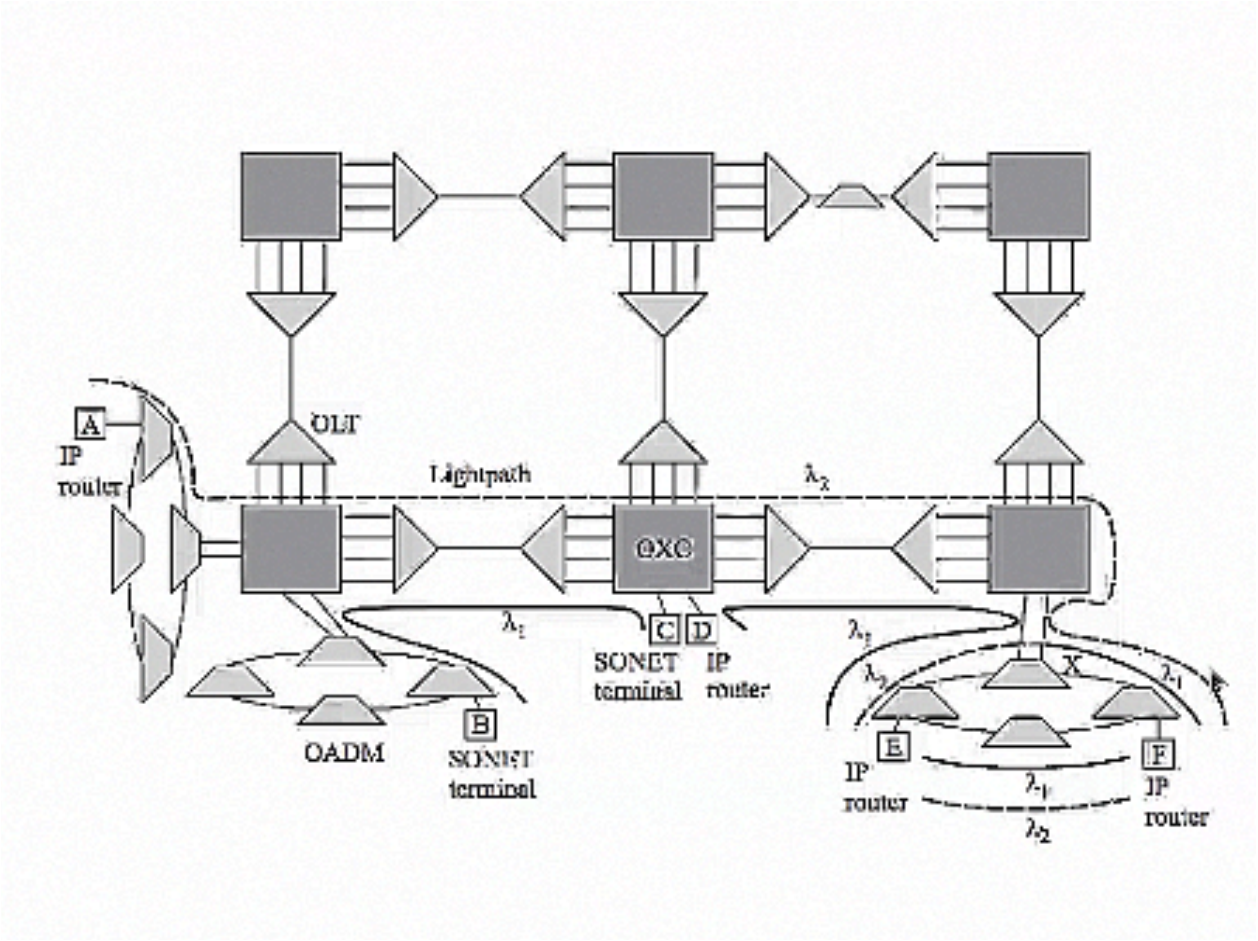
[2] it's the same thing thanks λ = c / f of the signal, with λ wavelength of the signal and c speed of light



Now we can switch directly in the optical domain, thanks to the color of the light. On each intermediate link λ of the lightpath can change.



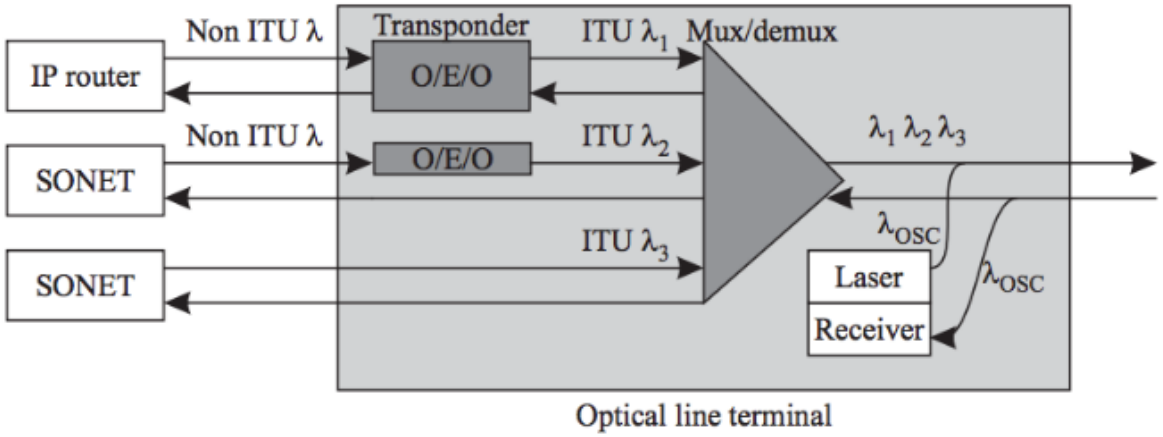
Representation of a Second Generation Optical Network:



The core network has:

1. Optical Line Terminal OLT: used at the ends of a p2p to multiplex and demultiplex wavelength. Is like a NIC, so there is a laser that generates the light ray. It's composed of 3 different elements:
    a. Transponder (O/E/O converter). A transponder does three things:
        i. It converts the signal into a suitable wavelength for the network. The OTN (Optical TN) can specify 1.55 micrometers as center for wavelength, while the signal is centered at 1,3 since it costs less to send it (it only needs a led), whereas I need lasers for 1.55. The OLT move the signal at 1.55
        ii. It adds OTN overhead to the client signal (OPU, ODU, OTU, FEC, …)
        iii. It monitors the BER: OLT is the point where the signal passes from Optical to Electronic so we have access to bits and we can monitor the BER.

    Transponders are the bulk cost in an OLT, so we want to have few of them.
    b. Wavelength multiplexer
    c. Optical Amplifier



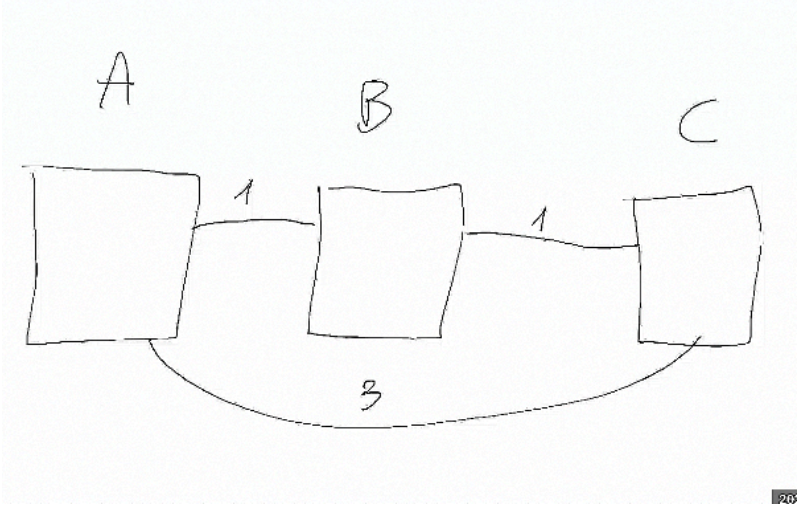
Non ITU lambda can't be sent in the network, we need to center it at 1.55. In the output, the signal has a specific lambda: λ_1, so we already have decided the color. This is because transponders have a specific color and they will always produce it. Actually nowadays we have tunable transponders so we can change the color.

SONET[3] doesn't need a transponder, because the transponder it's already integrated or the signal is already centered at 1.55.

For monitoring purposes there is an extra, dedicated, channel: optical supervisory channel (OSC), a specific lambda λ_{OSC} used to check the health status of the fiber. λ_{OSC} can't be used by the users.

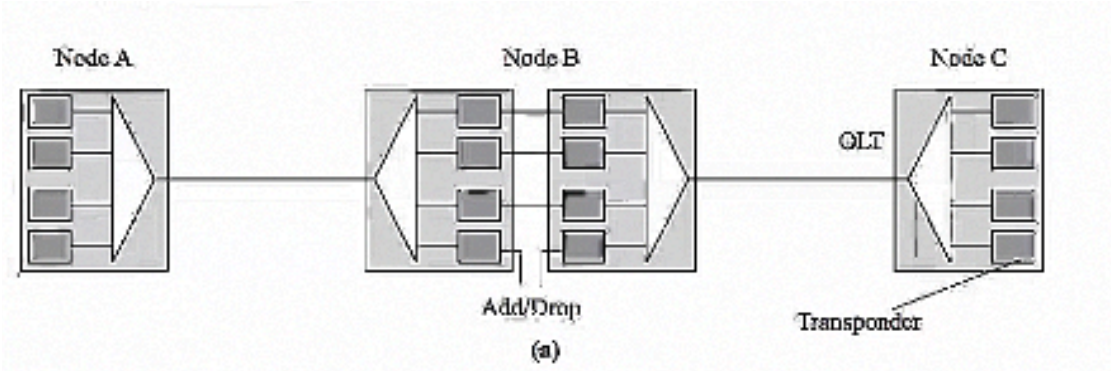
Here it is not reported but before sending the signal we amplify with an optical amplifier.

2. Optical add/drop multiplexers (OADM): cost-effective way for handling passthrough traffic in both metropolitan and long-haul net. Suppose you have 3 different nodes A, B and C. We want to connect logically A to C, with the following requirements: the capacity of AB and BC is 1, we want to pass 3 wavelengths from A to C.

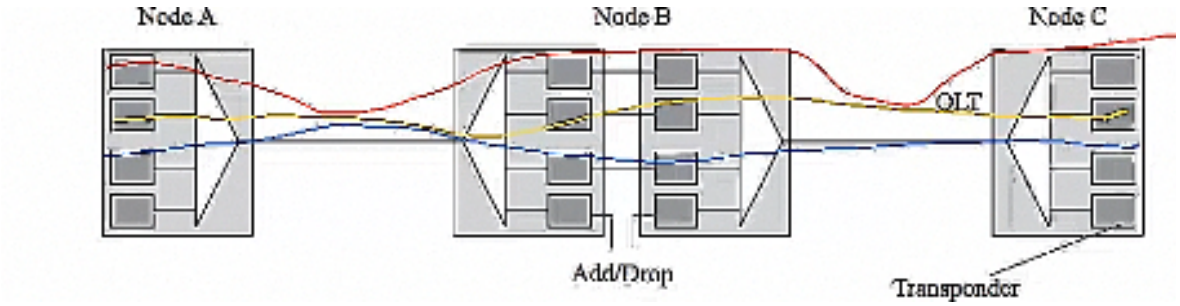


Using only OLTs:provide

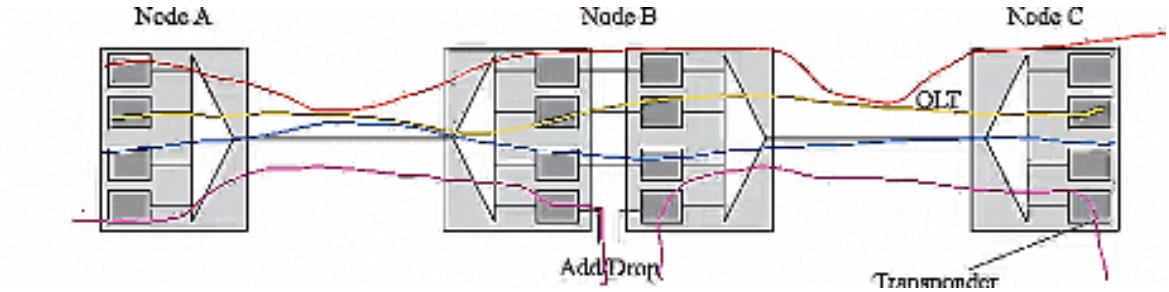
[3] Synchronous Optical NETworking



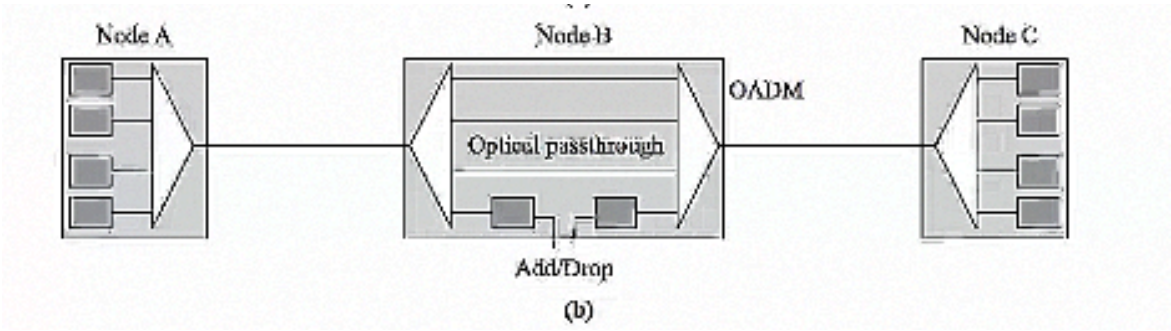
I send 4 lambdas, 3 of them are sent to C through B



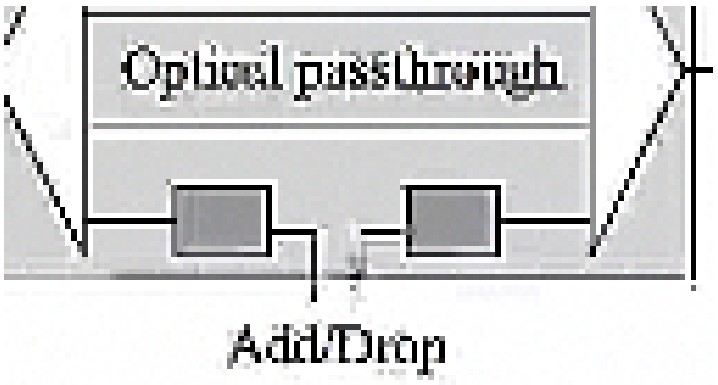
One of them is dropped locally in B to interconnect A with B and B with C



This way we are using 16 transponders, which are expensive. Since we are not dropping locally in B the 3 lambdas, we should make them pass through without doing O/E/O. For doing so we use a OADM, that uses only the necessary transponder.

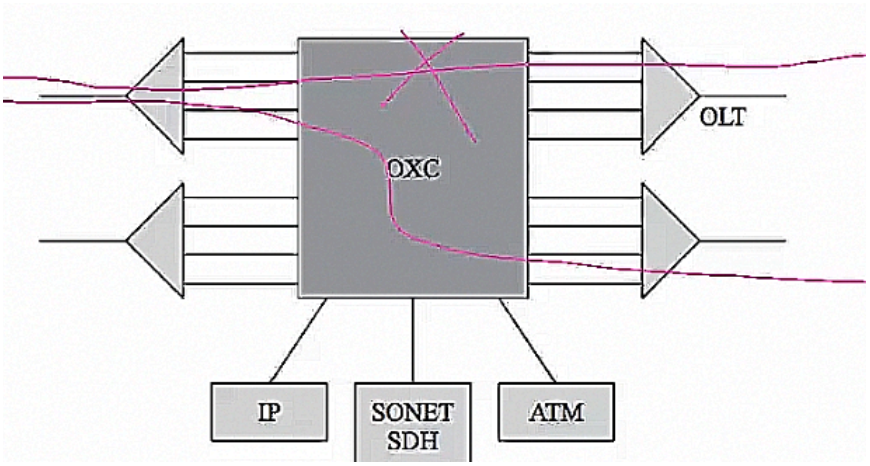


It's useful in ring topology, because most of the traffic is always passing through.



This OADM transponder is set to a specific wavelength, let's say green. So If they are built to drop green they will always drop green. If I want to drop another color or I want to let green pass through I need to buy a new OADM (assuming we are using the static version and not the tunable one).

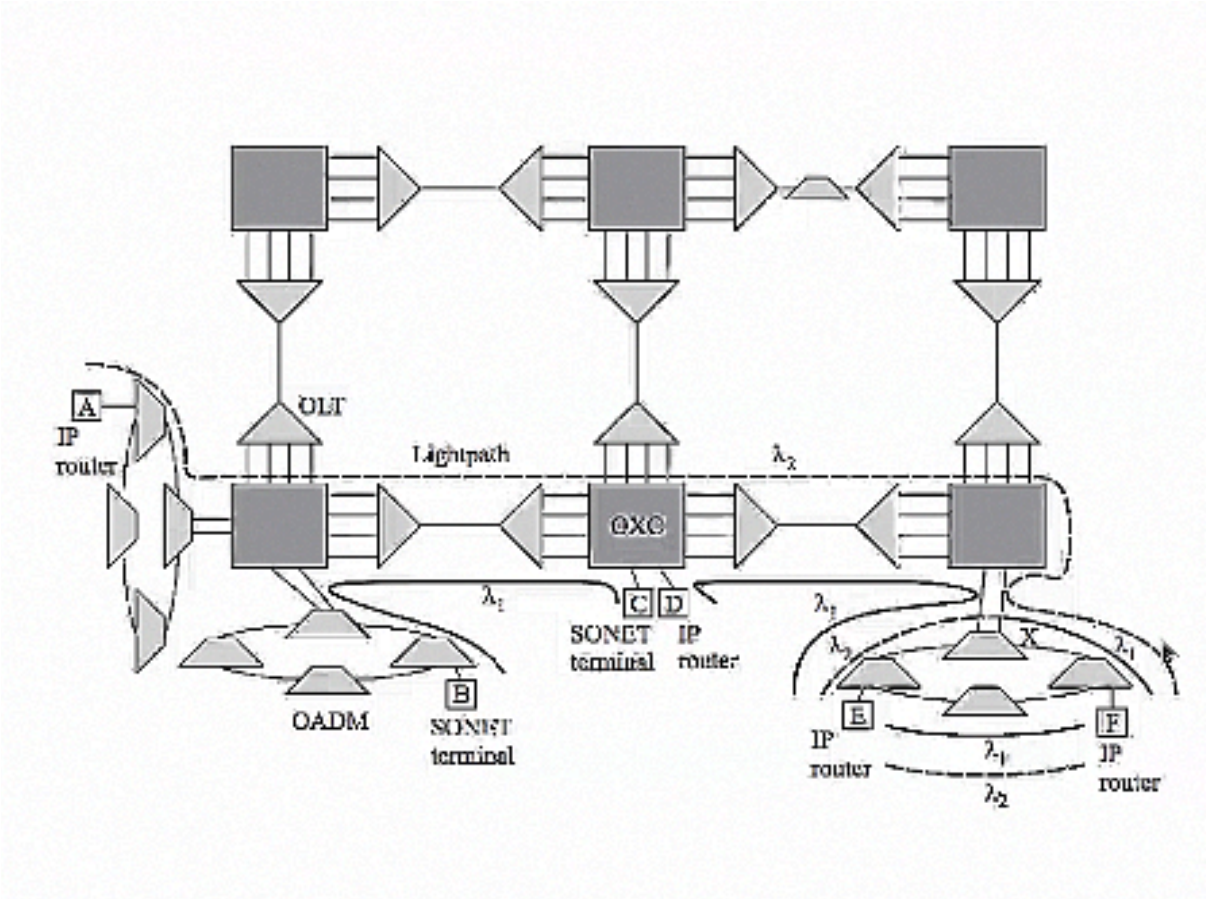
3. Optical Cross Connect (OXC): is similar to the OADM but on a bigger scale. Optical Cross Connect is like an IP router in the optical world: it has lights as input and it switches a signal in one of its multiple ports. It can perform switches dynamically: since it has to switch direction of a light ray, it uses mirrors that are reconfigurable in positions



In this case I said: I don't want purple to go up, pass it to the output below. The functions provided are:

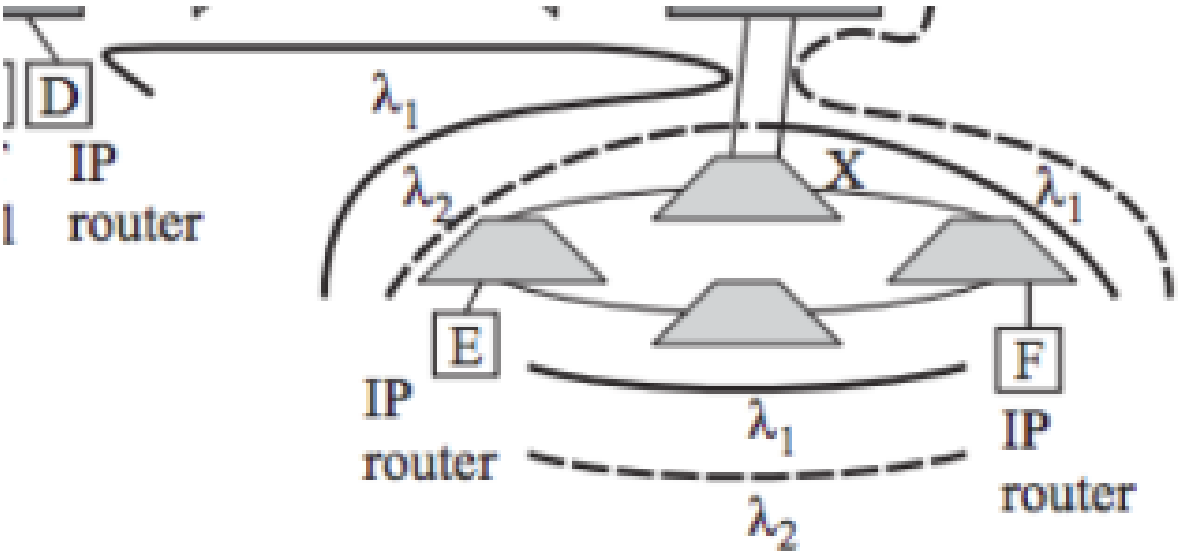
1. Automatic provisioning (fornitura) of lightpaths in a large network (no manual patch panel connections, so the switches are done automatically without human intervention)
2. Protection: reroute lightpaths after detecting a failure
3. Bit rate and Frame Format Transparency: switches with arbitrary bit rates and frame format
4. *Wavelength conversion*: change the wavelength before transmitting a signal. For instance I can take the red signal, convert it into an electronic signal and use a different color to resend it. For doing that we need to pass through the electrical domain.

5. Multiplexing and Grooming



At the border there are three rings. From this the users connect to the TN. The rest is the core part of the network, and we use a mesh topology to be able to find alternative paths between pairs of clients. Closer to the user we use OADM (the trapezoid) while in the core we use OXC cause we need to switch to a much larger scale.

We can't assign the same lambda to two users over the same link. For instance: D is communicating with E through λ_1. If we want E to communicate with F through the same link, we can't use λ_1, otherwise we will confuse the communication with the one between D and E, so we use λ_2. Arriving at X, where the link is no longer used by λ_1, we can apply wavelength conversion and use λ_1 to reach F.

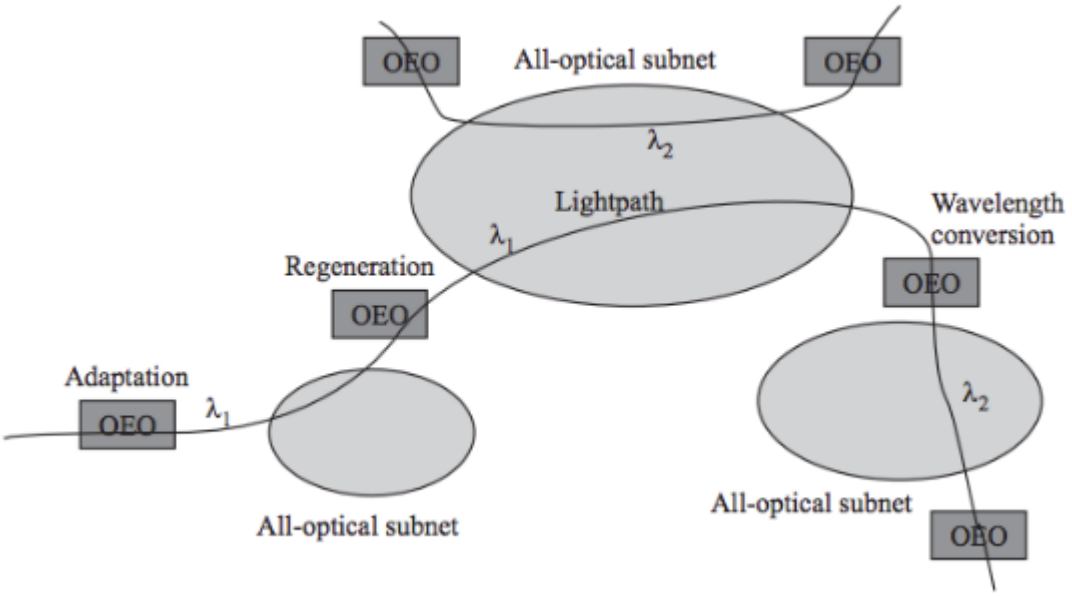


All-Optical Subnetworks

An optical layer is the physical layer of the TCP stack but since it performs routing and switching functions it implements a full stack of layers in layer 1 of the TCP stack. Why do we do that? Because the best option is to always stay in the optical layer, so don't use electronic devices. In this way I have better performance, less overhead (since I don't have to convert from O to E and vice versa) and the most transparent[4] network I can create.

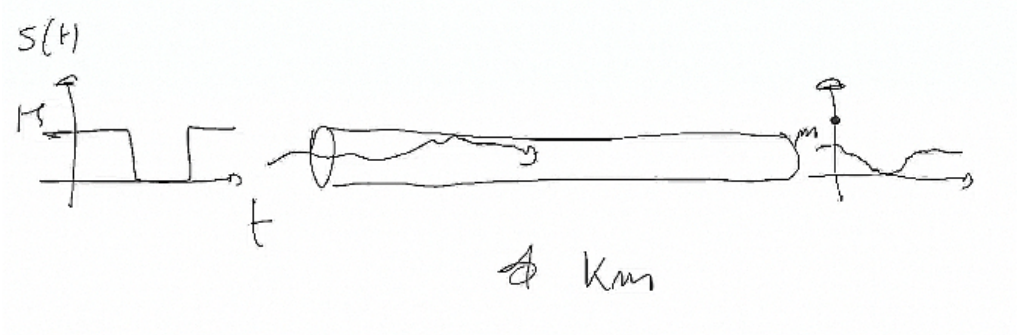
Since analog signals require higher SNR w.r.t digital ones, a completely Optical Network is hard to realize, so Optical Network almost always has electronics. What we end up with is All-optical subnetworks interconnected with points in which electronics are available, similarly to the IP situation in which we have subnetworks interconnected with routers.

[4] Transparency: we have transparency when we can use the same infrastructure to realize completely different networks. It's like having a bus, we know it can serve everyone, not only, for instance, workers, but also students.



Electronics is necessary to:

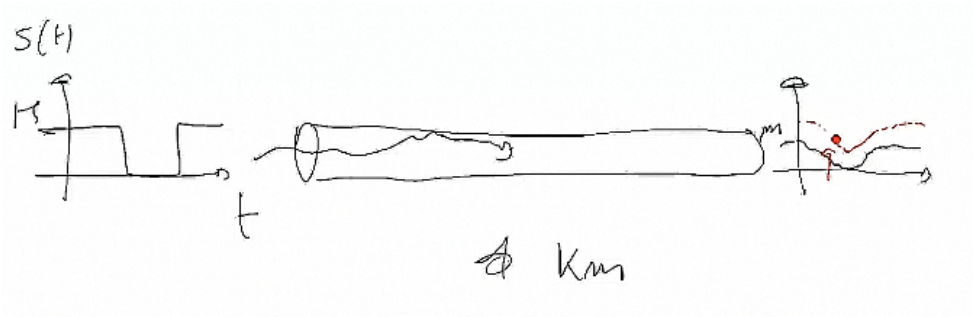
1. Adapt the signal entering the optical domain at the edge of the network. For instance we have to center the signal to 1.55 micrometers with a transponder.
2. Error correction using FEC
3. Measure BER
4. Perform Wavelength Conversion
5. Regenerate the signal digitally with:
    a. 1R regeneration (Optical Amplifier). When we inject a signal in the fiber we have a degraded signal in the output



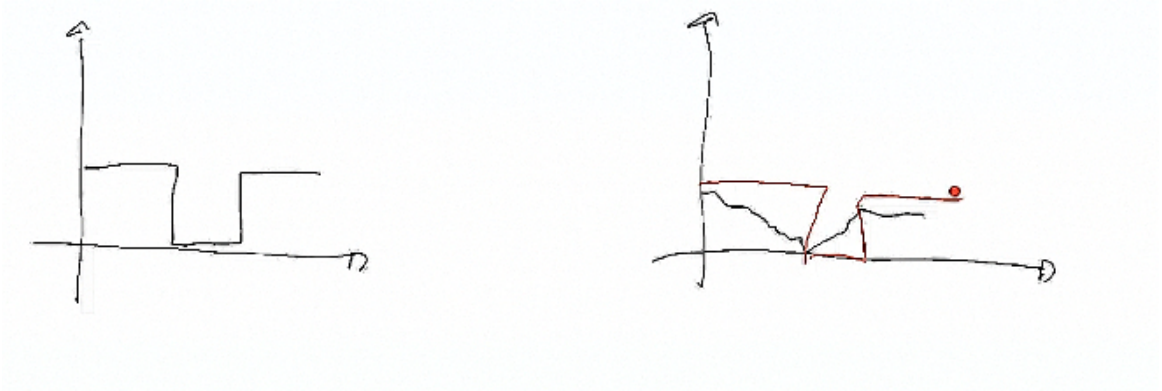
as you can see the signal has lose power during the propagation, the 1R adds power, moving the curve up.

PRO: supports analog signals, so it's transparent

CONS: It's not a real regeneration of the signal as you can see in the figure, since the shape is not the same.



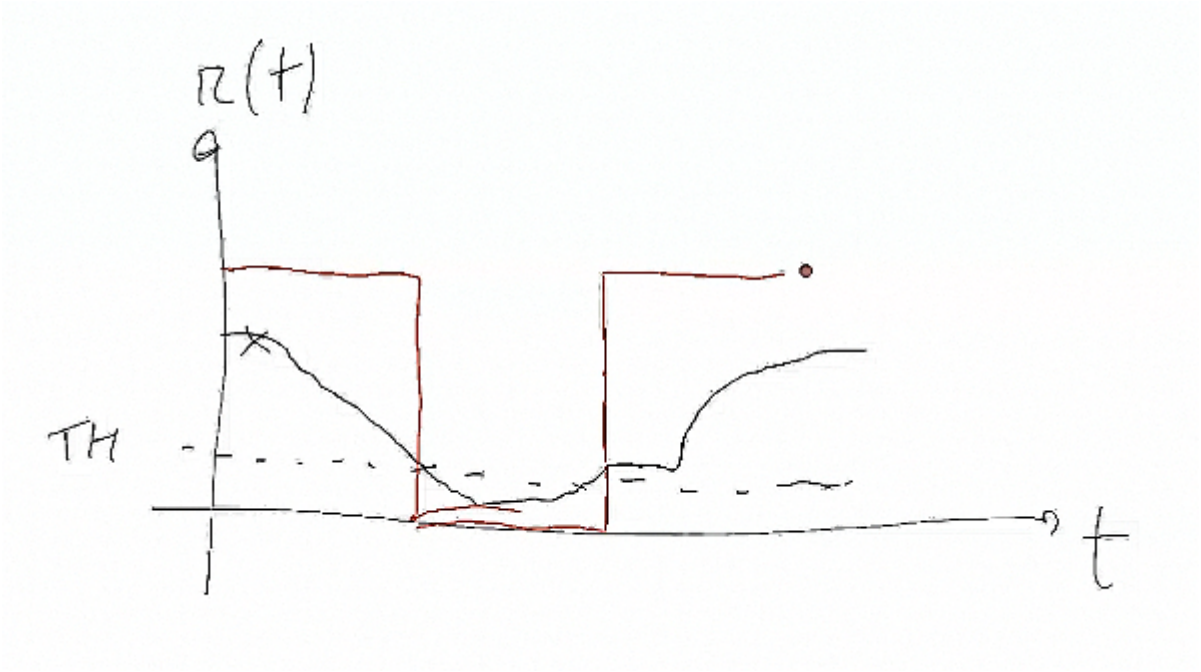
b. 2R regeneration (so adds power) with reshaping to the original shape.

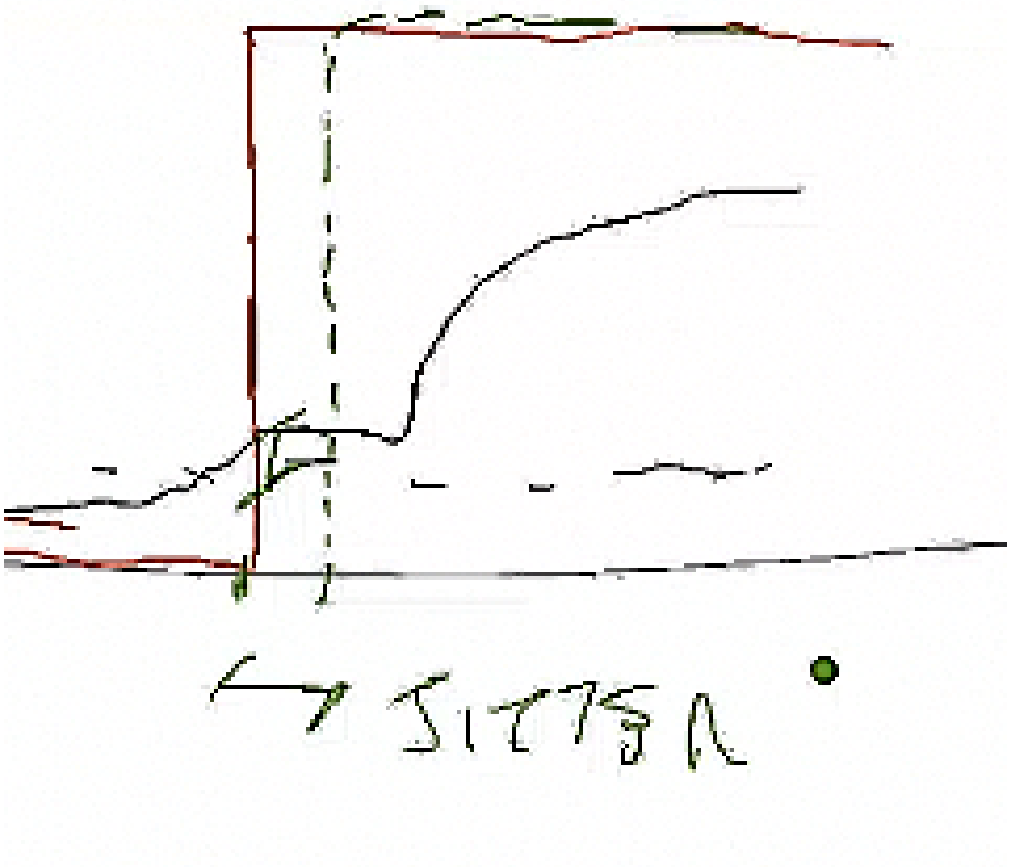


PRO: The 2R can regenerate 1 Mb, or 1 Gb it's not a problem, so it's transparent with regards to the bit rates, but it's not transparent with regards to the shape anymore: it can regenerate only square shape.

CONS: Jitter. Since it works according to a threshold (it decides if the received signal is 1 or 0 by comparing the value of the signal with the threTshold) it could happen for noise that a signal overcomes the threshold just

for a small quantity and the signal is misinterpreted.



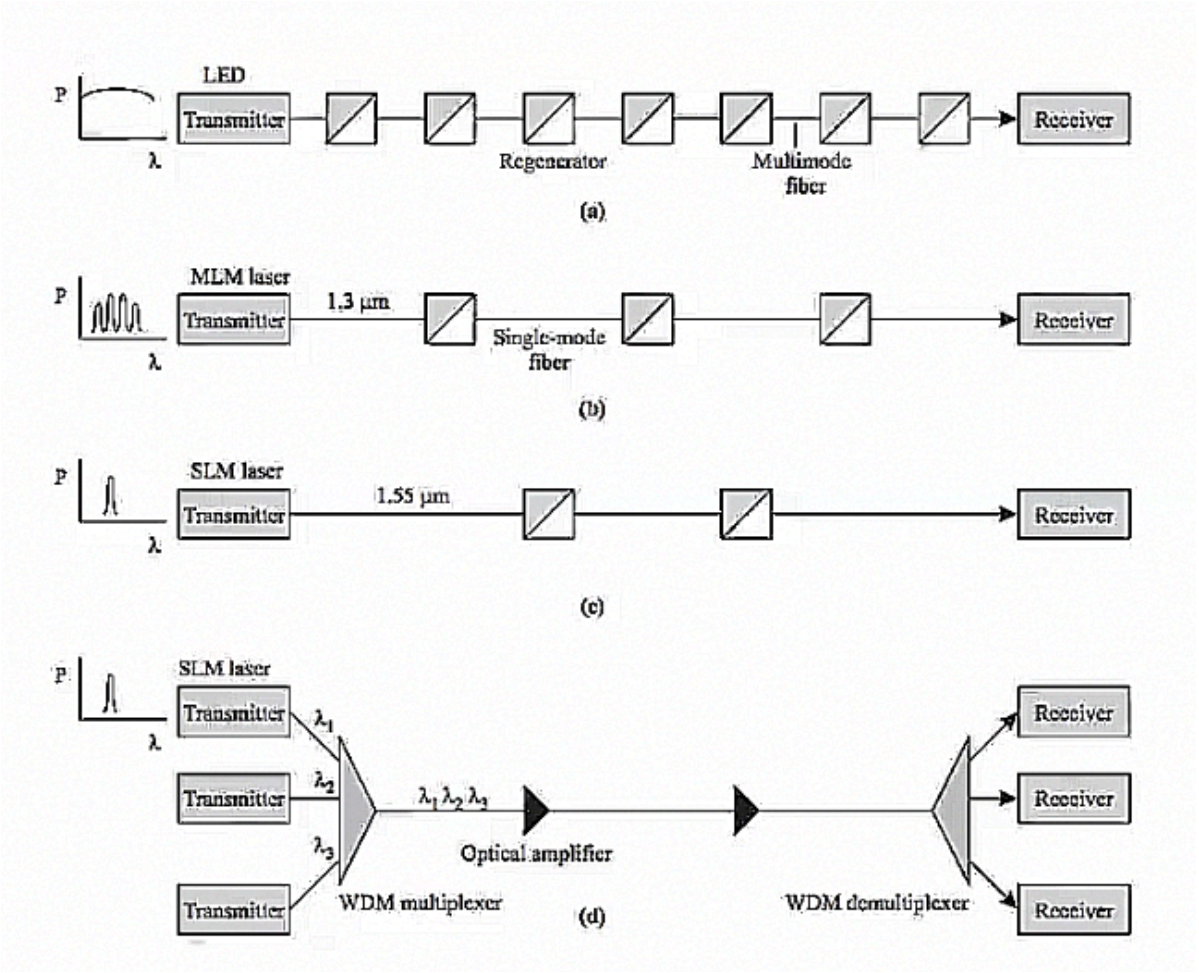


c. 3R regeneration with reshaping and retiming. Is the transponder.

PRO: this produces a “fresh” copy of the signal

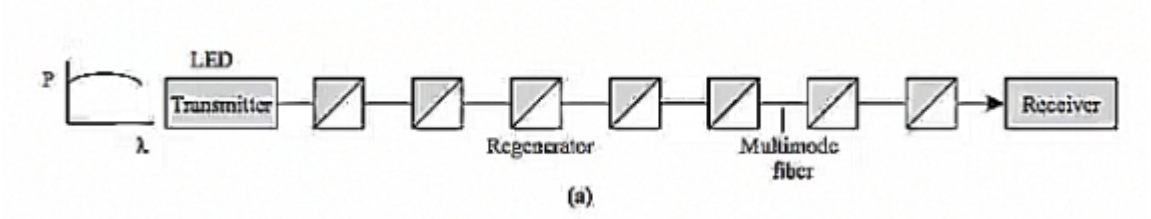
CONS: it's not transparent at all since It exists to transpond to a specific client signal. I can't use it for a different client, because a transponder has a specific bit rate.

Network Evolution

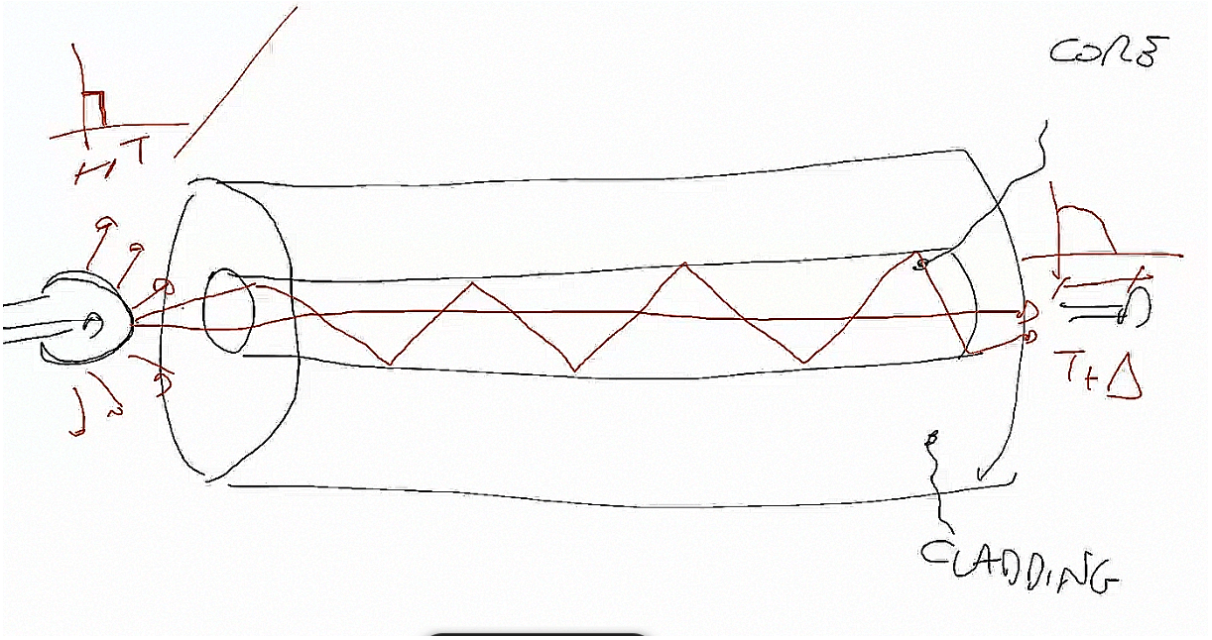


As we go down the years are passing and we become more transparent so we use less electronics.

a. First case:

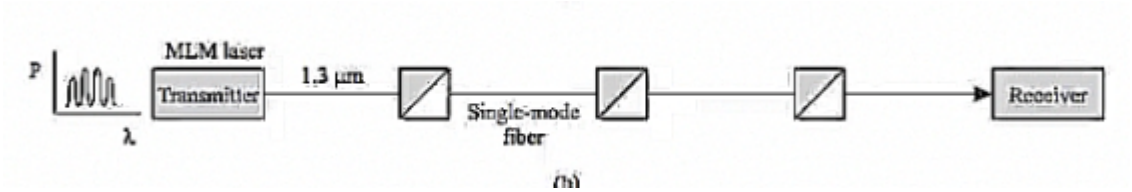


We were using LEDs on multimode fiber. The problem was *Multimode Dispersion*: we needed all these regenerators because the signal transmitted through multimode fiber is composed like that:

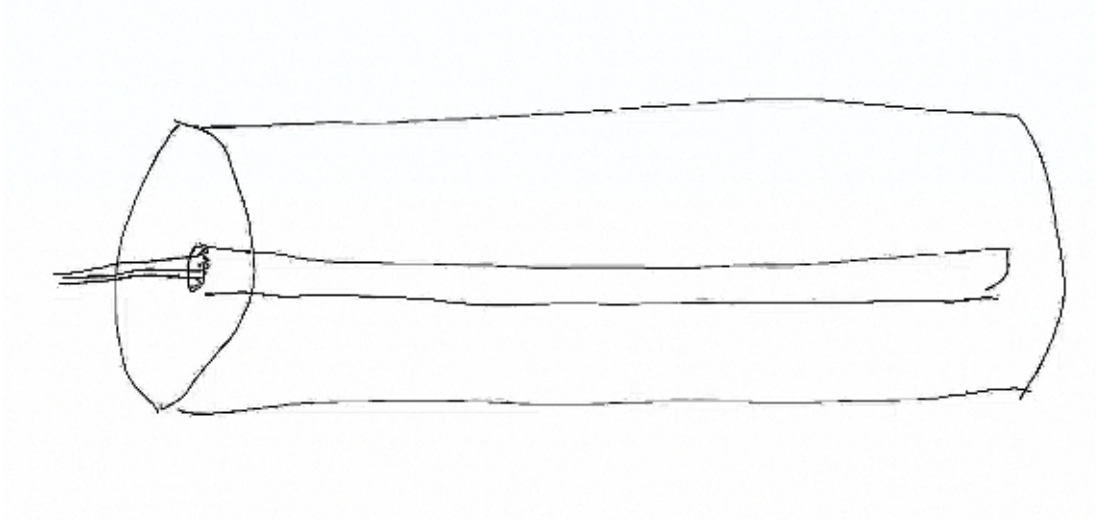


The problem is that when the light bulb is on and the light rays are going through the channel they have different orientation, as shown in the picture. So there is a straight ray going directly and in the fastest way through the channel, then obliques rays that are going to reach the output later. This will degrade the signal, as shown at the end. To solve that, we need to regenerate the signal.

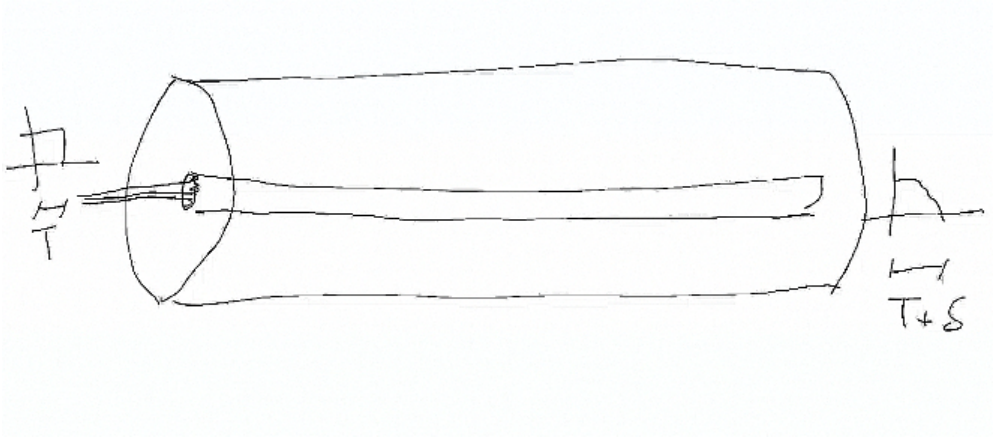
b. A system using MLM laser over a single-mode fiber in 1.3 micrometer band to overcome intermodal dispersion in multimode fiber.



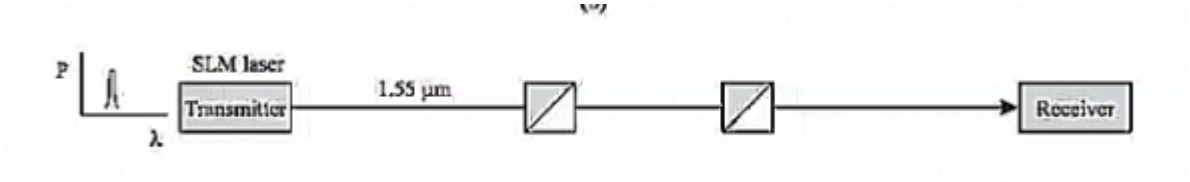
We can use single mode fiber that has a very thin core, this way there are still multiple rays that enter the core but since the core is thin the paths of the rays are almost the same.



We need 1.3 micrometer window but the attenuation at this center is much higher so we still need regeneration. We have another problem too, *Chromatic Dispersion*: even if we use single-mode fiber, it's not actually true that the light rays have the same speed, because it also depends on wavelength. So we still have degradation due to the delay in some rays that reach the output at time T + s instead of T.

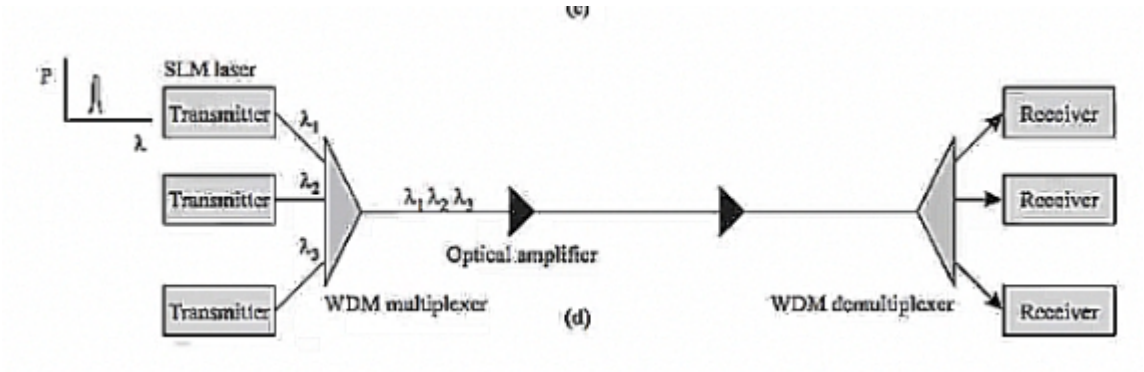


c. A system using a 1.55 micrometer band for lower loss, and SLM laser to overcome chromatic dispersion limits. With 1.55 micrometers the attenuation is lower so we reduce the regenerators used and thanks to SLM laser we solve the problem of chromatic dispersion



In fact, an SLM laser has a much more precise source of light in terms of output color, so the light ray transmits just one color. Until here, the systems were 1st generation optical networks, so we needed to switch in the electronic domain.

d. A 2nd generation WDM system with optical amplifiers instead of regenerators. Different lasers share the same p2p link through a multiplexer that sends a single light ray containing all the several colors (lambdas). The degradation is so small that amplification can be performed directly in the optical domain. This is the most transparent of the systems.



The P-lambda curves on the left indicate the power spectrum of the signal transmitted.

Question:

Which one of the following devices cannot be found inside an optical subnetwork?

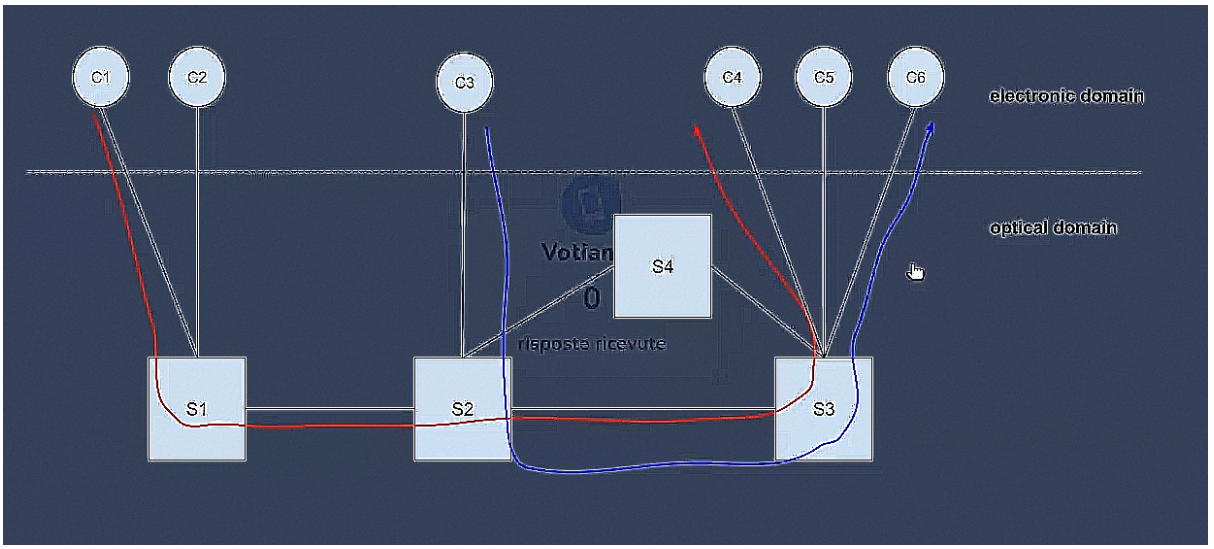
a. OLT
b. OXC
c. Optical Amplifier
d. OADM

Answer: between these elements the only one that has transponders is the OLT, is the device we use to pass through an optical subnet to another and the one that performs the O/E/O conversion, so this is the one that can't be found inside an optical subnetwork

Question:

Consider the All Optical Net shown in the figure, where each client is equipped with a single laser (red or blue), and whose switching elements can handle only these two wavelengths. Answer the following questions:

1. Can a lightpath be established between clients C2 and C5?



If so, what color can be used for this new connection?

2. *Now assume that two new clients (C7 and C8) are connected at the net, through S1 and S4 respectively. Can the network support the creation of a lightpath between these two?*

Answer:

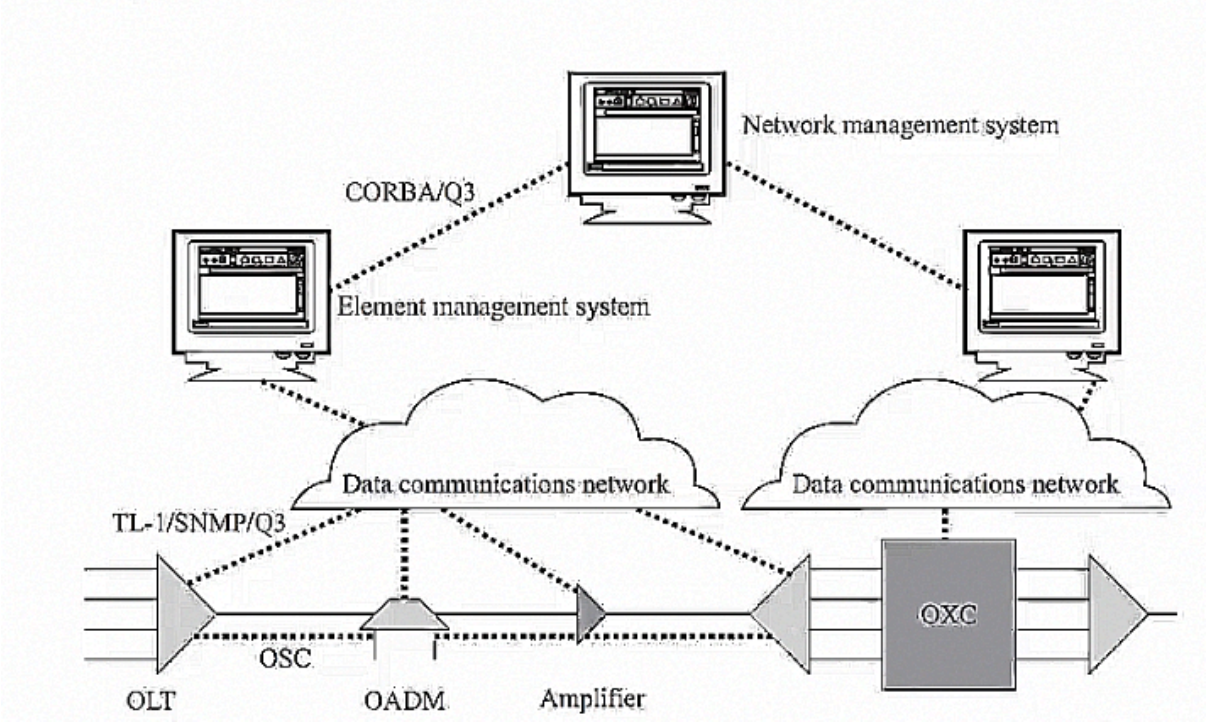
1. Yes, we can establish a lightpath between C2 and C5, using blue since a red lightpath is already present in S1 -> S2. Then we have to pass through S2 -> S4 and S4 -> S5 since the S2 -> S3 has a blue lightpath already.
2. No, the network can't support the creation of a lightpath between C7 and C8 since S1 -> S2 is already covered by both red and blue lightpath, which are the only admissible by the network. We have to buy a new transponder with a new color.

Control and Management of an Optical Network:

In an general network, we have:

- Data plane: where user data are sent, the optical part in our case.
- Control plane: performs control functions like routing
- Management plane: set of functions that are needed to manage the devices

As the net grows we need a framework that does control and management automatically. Optical networks are managed by a central, hierarchical system that is obviously electronic.

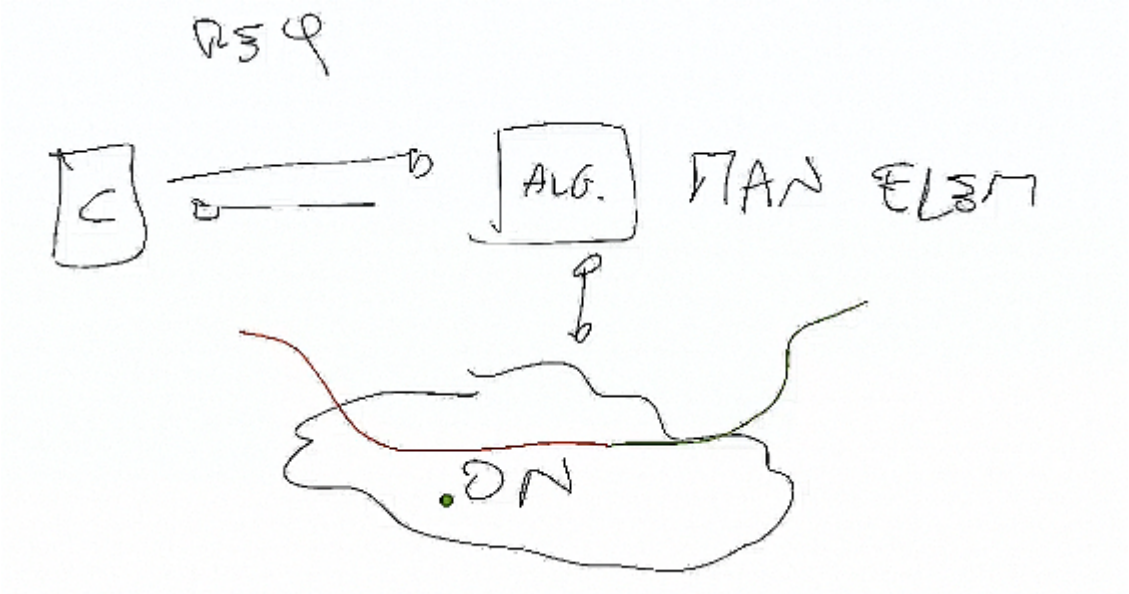


A client arrives and makes a request, in the request it specifies:

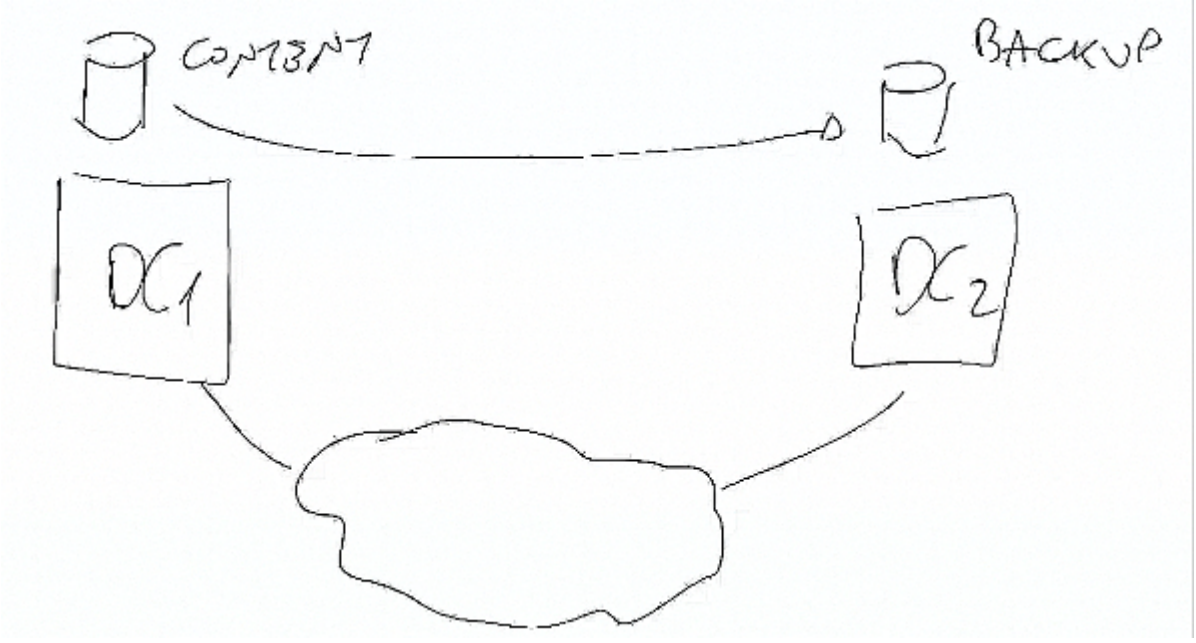
- the endpoints to interconnect
- the amount of bandwidth

- BER
- level of protection against failures
- requirements related to jitter and delay
- …

The management element performs an algorithm and determines the light path.



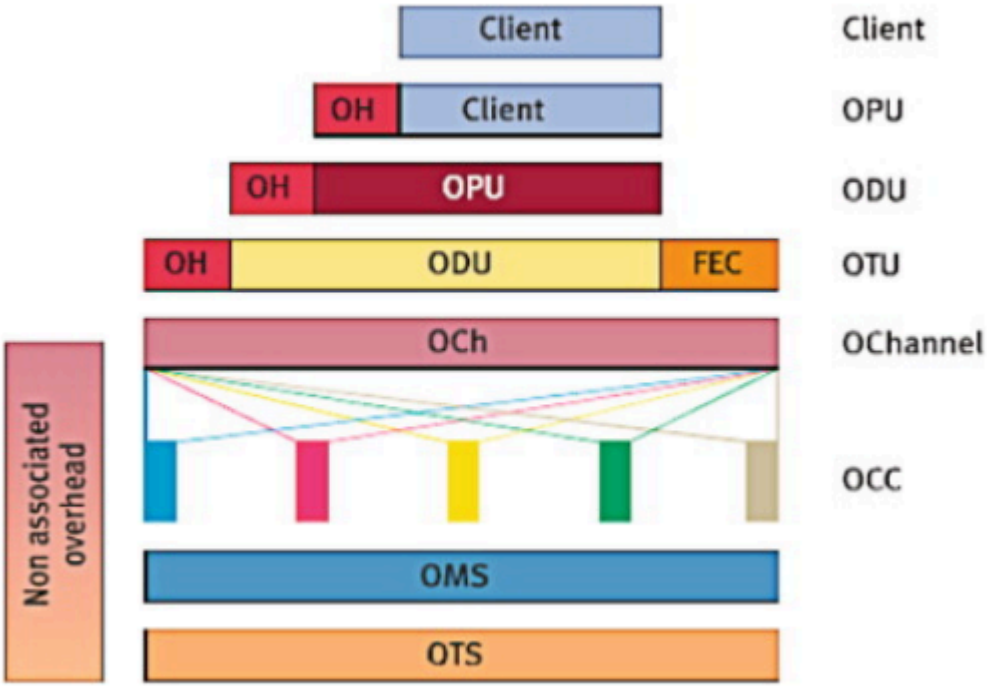
This procedure is time consuming, let's make an example to show the time consumption: we want to provide a backup of the content



The cloud provider asks for a lightpath that interconnects the 2 DC (data center) just for the time needed to create the backup, so 1 day, maybe some hours. It's a big challenge because the management infrastructure requires a lot of time to set up, so if you want to use the lightpath just for a few hours it may not be worth the time spent setupping. So basically: it works fine as long as lightpaths are set up fairly infrequently and remain nailed down for long periods of time, but what if this is not the case?

Well, we can think of performing some control and management (such as failure identification) functions in a decentralized manner directly by the optical network.

To delineate management functions it is useful to subdivide into electrical and optical domains and into several sublayers for each domain:



For the Optical Domain:

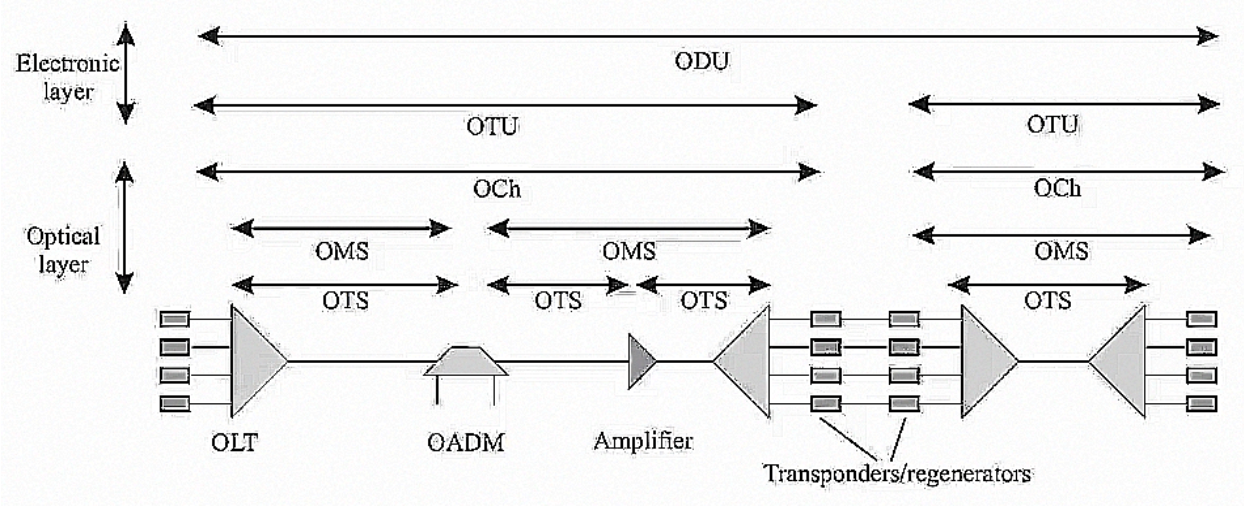
1. Optical Transmission Section Layer (OTS): on the link between two optical amplifiers, so we see every single fiber pair.
2. Optical Multiplex Section Layer (OMS): on the link between OLTs or OADMs
3. Optical Channel Layer (OCh): for end-to-end routing of the lightpaths

For the Electronic Domain:

1. Optical Transport Unit (OTU): provide communication channel between the endpoints of the optical connection
2. Optical Data Unit (ODU): as OTU but at a higher layer. Includes:
3. Optical Payload Unit (OPU) sublayer

So, starting on top with the client, its signal is encapsulated in a data unit and we add the OPU overhead. Then we have ODU layer, and OTU layer, where FEC is used to recover from binary errors. Passing from OTU to the OCh we are passing from the electronic domain to the optical one. This passage here is performed by the transponder. Also the adding of headers is done by transponders.

In the OCh I can still distinguish the users thanks to the colors. After this layer we have the OMS in which all the signals are passed through a multiplexed and they are mixed in a single light ray and I can't distinguish between users anymore. It's like passing from IP to Ethernet. An Ethernet frame can bring packets belonging to the communication between two hosts or other. Finally the signal is transmitted in the cable: OTS

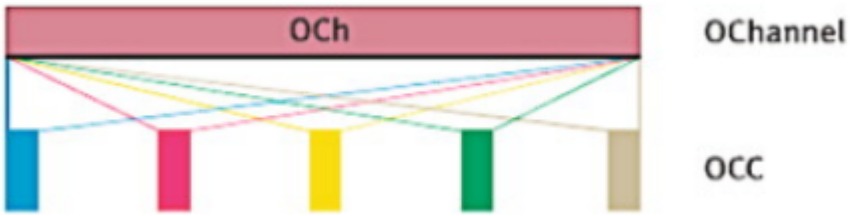


As shown in the last image, the ODU header is added at the start and decapsulated at the end. Just like IP routers don't decapsulate TCP header, that is decapsulated only at the end, in the terminal of destination. OTU is decapsulated every time we pass from optical to electronic, so whenever we meet a transponder, same for Och.

Question:

In what layer of the OTN hierarchy do we see the access to the signal carried by a single wavelength?

Answer: in the OCh layer, in which we still can distinguish between users. After that, in the OMS layer, all the wavelengths are mixed in a single light ray.



Alarm Management:

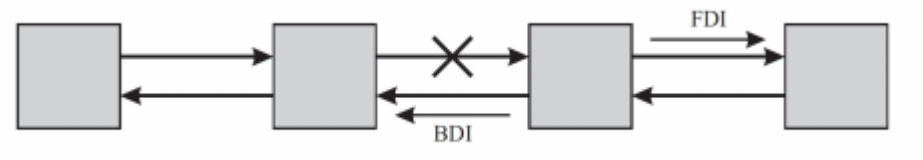
Let's say I have a network and a link fails, what do we do? We create an alternative lightpath, until the fault is repaired, but I need a way to detect failures and then to restore the service? These two are performed by Management Element. We will not see how to restore it, but just how to detect it.

How to detect failures:

1. BER. We can detect failures monitoring the BER continuously, but we only have access to the binary view in transponders, at the edge of an optical subnet. We use OTU and ODU overhead for BER monitoring.
2. Optical Trace. When we are not at the edge, we have an unique identifier for each lightpath, called Optical Trace, that enables the management system to get information about the 2 ends

These are the 2 tools we use to identify the failed component. How can we identify the root cause of the problem?

Let’s make an example:

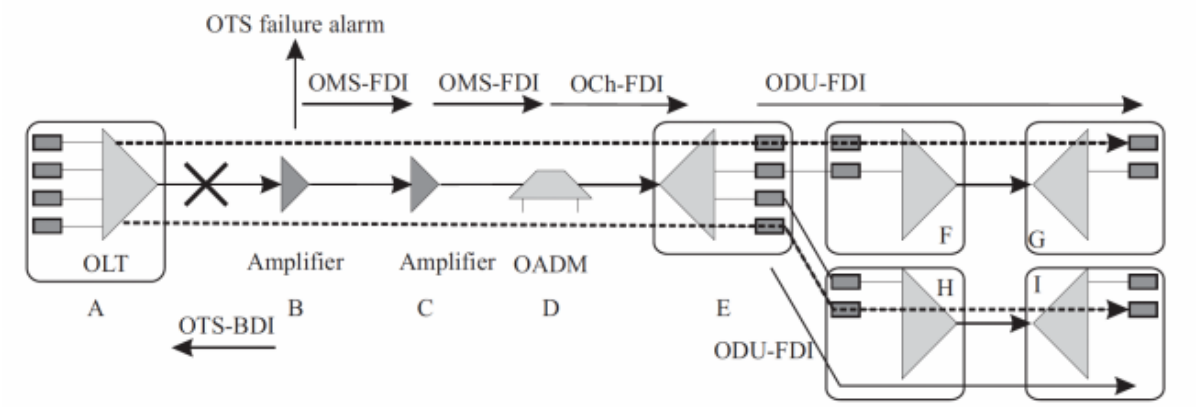


Let’s assume we have 32 lightpaths on each link. When the marked link fails, 32 * 4 + 1 alarms are raised. 32 * 4 alarms are raised because each node, that monitors continuously its lightpath, sees that they failed (so they don’t see the link failing, they just see the side-effect: the lightpaths failing), +1 alarm because the 2nd node sees the physical link failing. I have to suppress the redundant alarms and leave only the alarms that help me identify the root cause. We suppress the alarms with DI (Defect Indicator):

1. BDI: sent in the opposite direction with respect to the failed link
2. FDI: sent in the same direction.

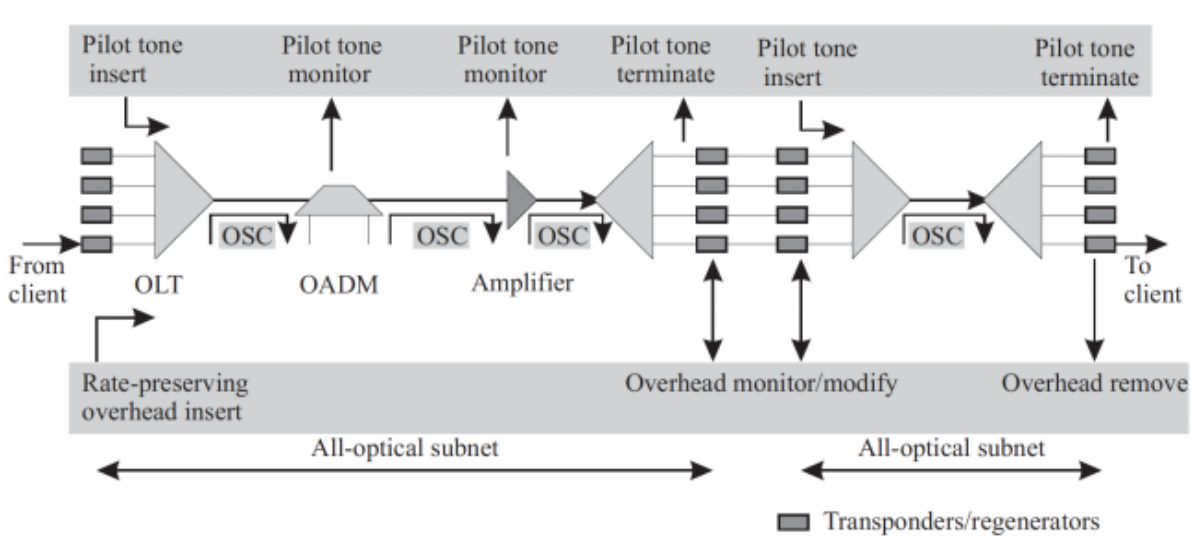
So the procedure is this: when there is a problem everyone starts raising alarms. After that the third node, that is attached to the failed fiber, starts sending FDI and BDI messages. When a node receives FDI or BDI immediately stops raising the alarm. But why is the third node sending FDI and BDI? Because it is one of the two nodes attached to the failed fiber, and the other one (the 2nd node) can't send FDI because the forward link is broken.

At the end only the third node, the one that sent FDI and BDI, will raise alarms, so we identified the node attached to the link that is broken.



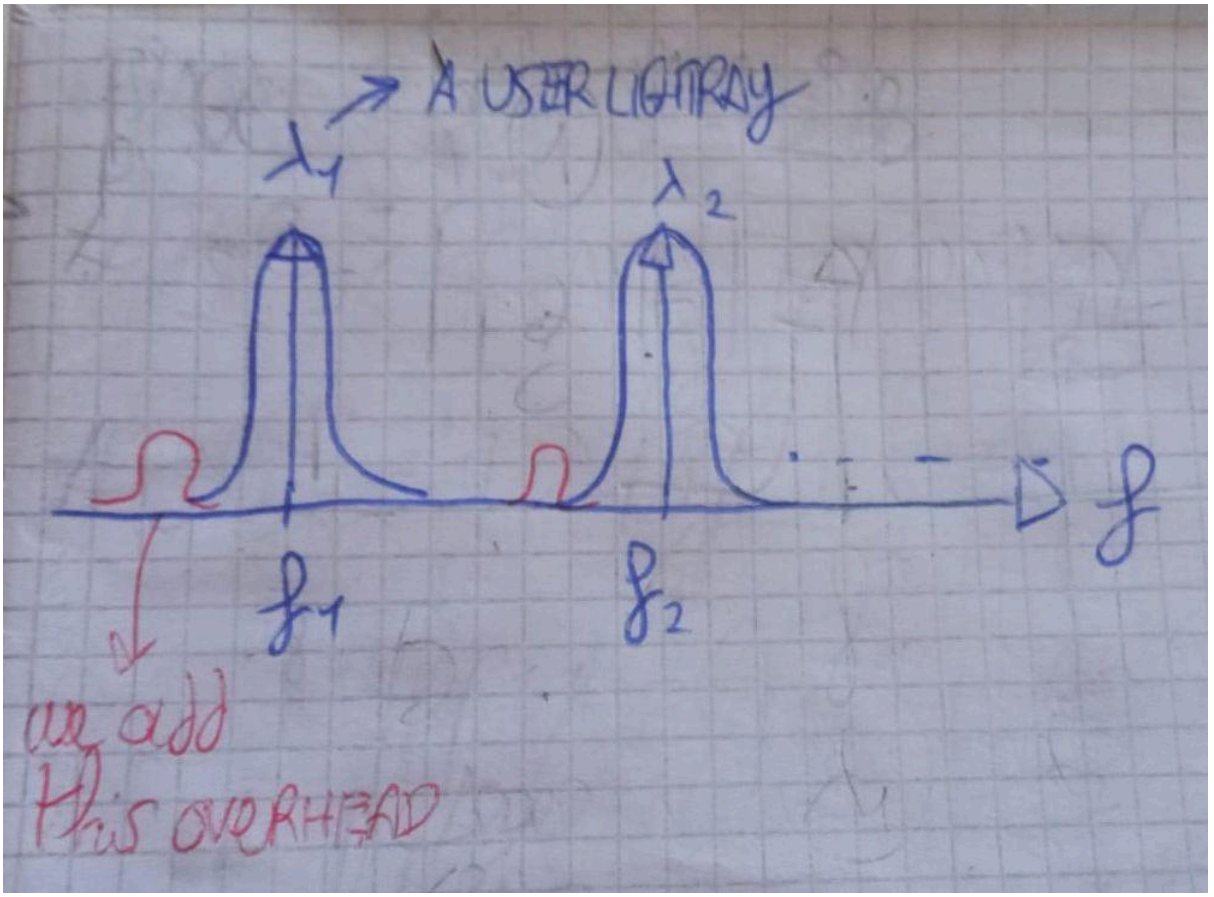
As shown in the image, FDI and BDI messages use optical layer overhead and are sent at different sublayers of the electronic and optical layers. But how can we add overhead in the optical layer?

Optical Layer Overhead:



As shown in the image, we have two overheads:

1. Pilot Tone



through modulation technique, pilot tone is added. It's a small overhead. Is in Och because is an overhead in the optical domain and, as it's shown in the image, we have a single lambda, not the mix of all the lambdas in a single ray (so it can't be OMS or OTS). The device that adds the pilot tone is the transponder. Since it's at Och layer, we can insert and remove it only at the edge of the All-Optical Subnet, where there is an OLT.

2. OSC (Optical Supervisory Channel): it's overhead for OMS and OTS. OSC is a special lambda dedicated to monitoring as we introduced in a previous lecture.

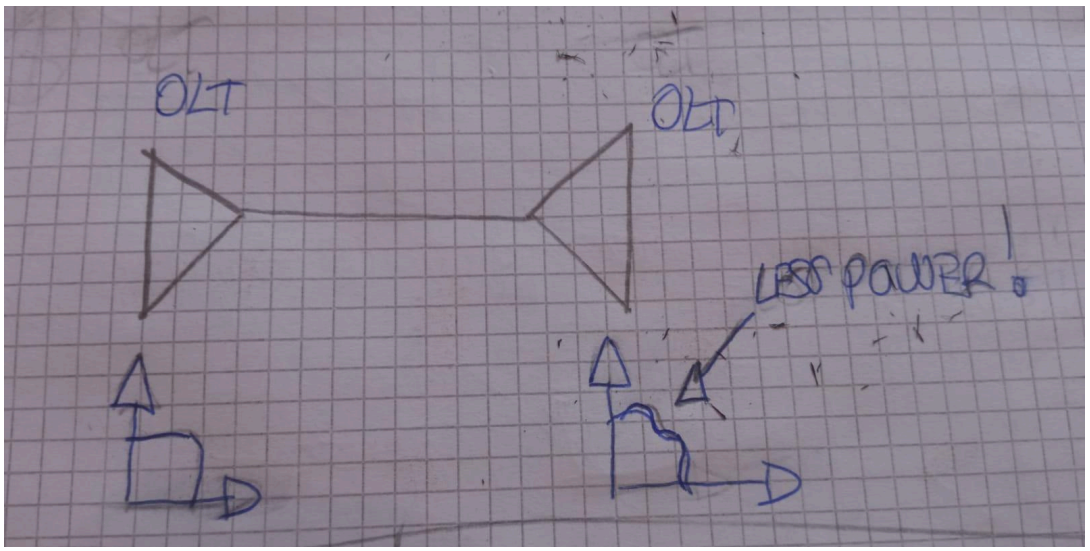
Here are some applications that uses OSC or Pilot Tone overhead:

	All-Optical Subnet		End-to-End
Application	OSC	Pilot Tone	Rate-Preserving
Trace	OTS OCh	OCh	OTU ODU
DIs	OTS OMS OCh	None	OTU ODU
Performance monitoring	None	Optical power	BER
Client signal compatibility	Any	Any	Any

- Trace:
    - With OSC: we will create a trace for every fiber link (OTS overhead). In this case we can't assign an identifier to a lightpath because it's much longer than a single fiber link.
    - With Pilot Tone: if we want to assign an identifier to an entire lightpath, since with Och header we can define a trace for an entire optical channel. In this case, in the intermediate nodes we can just monitor the pilot tone, and we can change the information only in OLTs.
    - OTU/ODU overhead in the electrical layer.

- DIs (Defect indicators): so FDI and BDI. We can't use pilot tone because pilot tones are added in the OLT, if an intermediate link fails I have no possibility of adding the info of the failing. We have to use:
    - OSC in OTS, OMS or Och layer in the optical domain
    - OTU/ODU in the electronic domain.

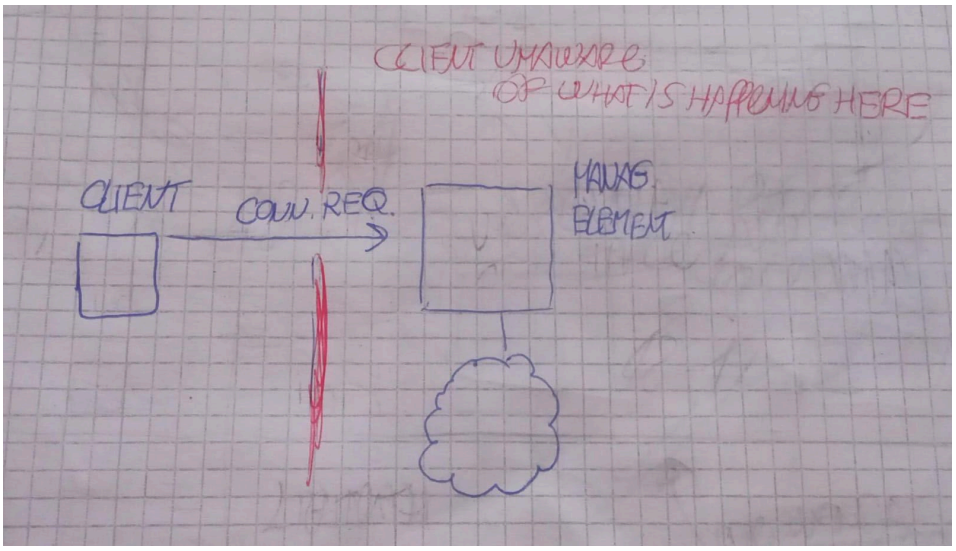
- To Monitor Performance:
    - OTU and ODU to monitor BER in the electrical domain.
    - With Pilot Tone, to monitor optical power. What does it mean to monitor optical power? It means this:



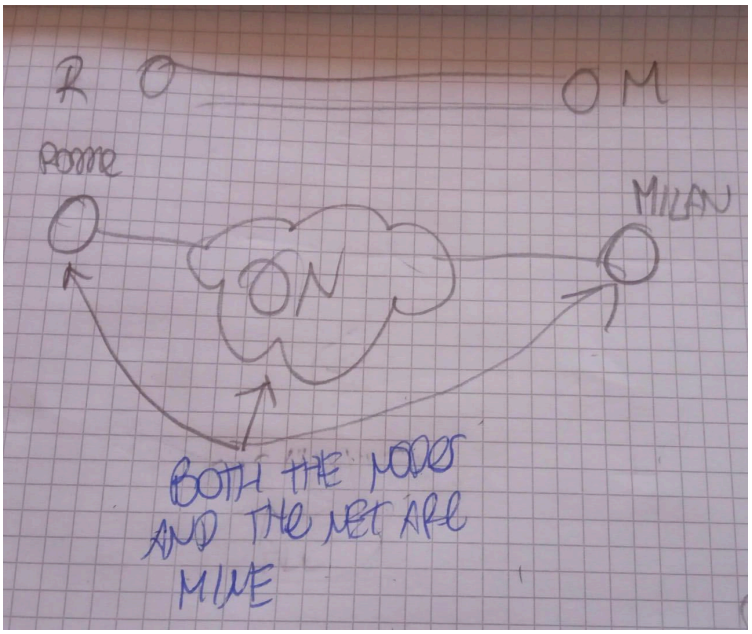
But when I arrive at the OLT on the right, I don't know the initial shape of the signal, I just see the distorted version. So how can I compare the two signals to check the loss of power if I don't know one of the two signals I have to compare? What I can do is to compare a signal whose shape is known (such as the pilot tone) with the same signal obtained after the signal went through the link, to check if it has less power.

Connection Management:
Deals with setting up connections and keeping track of them, taking them down. We have two approaches:

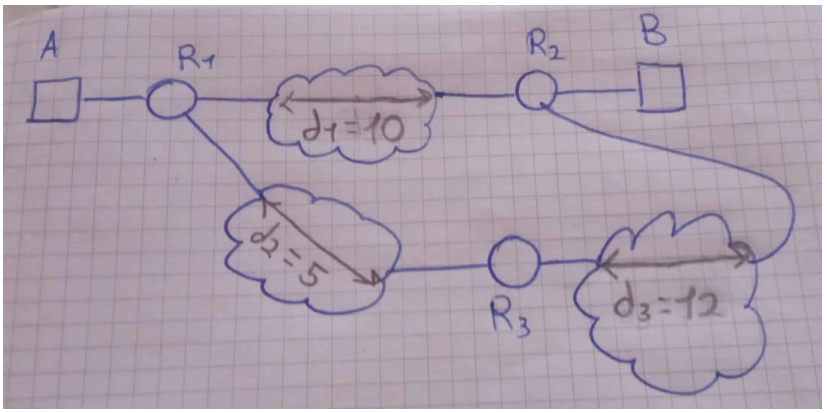
1. client-server model. In this mode the client is unaware of what's happening in the optical network and has no control of what the provider of the service he asked for is doing. However, it's more transparent and less costful and when the connection is asked infrequently and I keep it for long time is preferred



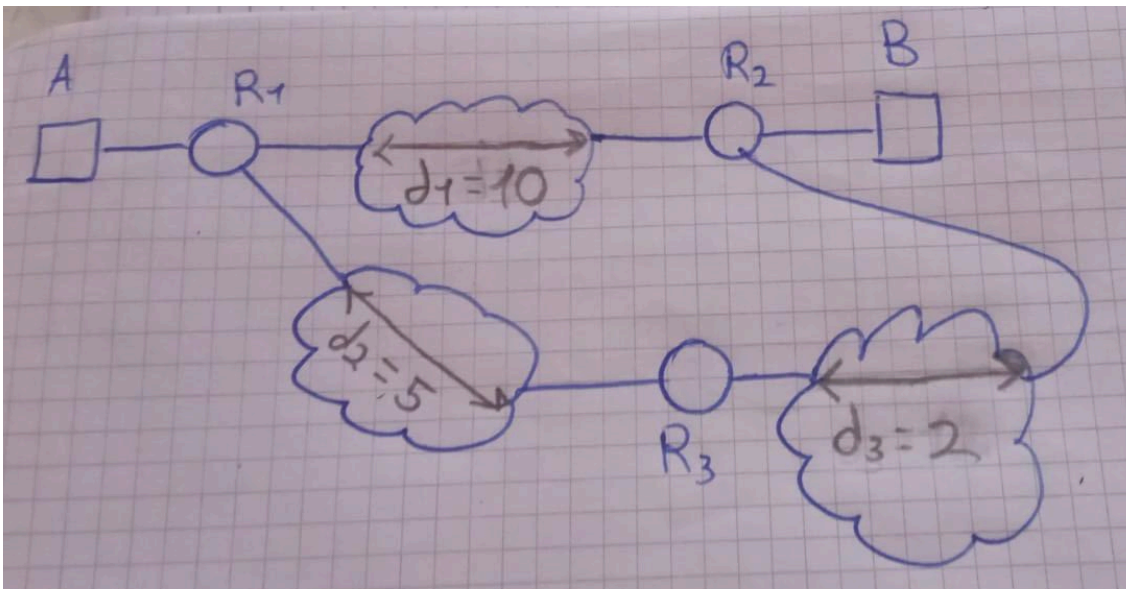
2. peer model: tight coupling between the client and the optical layers. So the network primarily serves a single client. More flexible, since there is a cooperation between the Control Plane of the client and the optical network.



Let's make an example:
First case: the owner of R1, R2, R3 is buying the optical net services from a provider. The provider offers the lightpath with delay d1 = 10. The other is d2 = 5 and d3 = 12. The optimal path is also the shortest one, through d1.



Second case: the owner of R1, R2 and R3 owns the network too: let's suppose that d3 becomes = 3. If I own the network I can switch the path from the shortest one to d1 to d2 + d3, which has less delay.



For doing so the router needs to know the topology. We use OSPF (routing protocol used in IP net that allows a specific node to be aware of the topology) for that. In this distributed approach each node must know the net topology, so we use OSPF.

Multiprotocol Label Switching (MPLS):

9/10 and 16/10

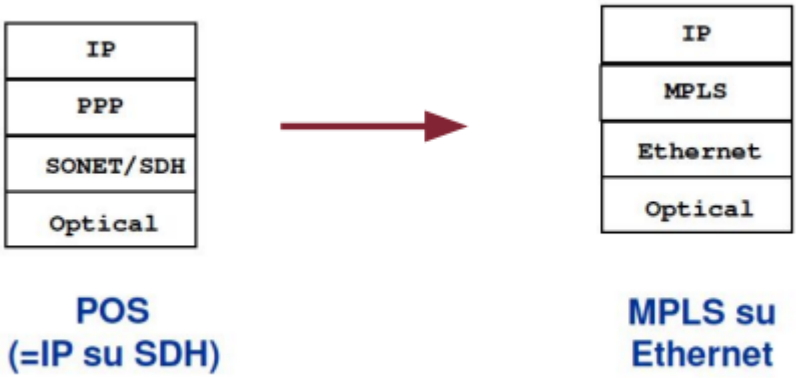
End users buy services from Layer 3 providers. The Layer 3 (IP network) is realized on top of an optical network and each link is realized with a lightpath. The layer 3 provider buys the lightpaths from the carrier.

A L3 provider should support:

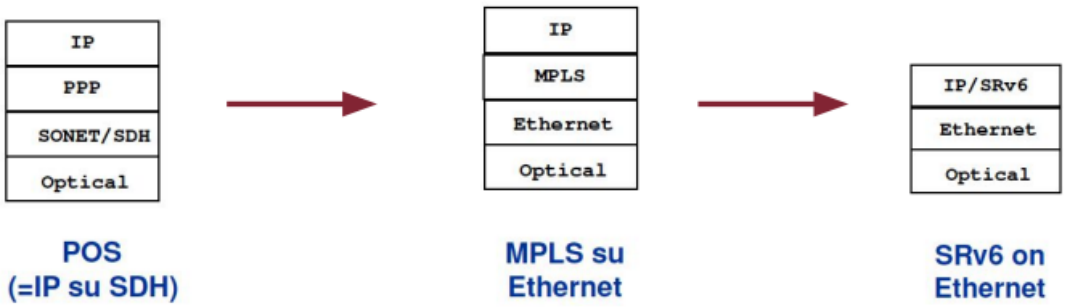
1. Traffic Engineering
2. VPN
3. Protection against failure and Restoration mechanisms

Connection-Oriented Packet Switching, so MPLS, enhances these three benefits that a provider can provide. It's Multi Protocol because we can encapsulate it in whatever protocol we prefer, not only IP (although is the main protocol used above it), and is independent from the underlying technology (SDH, Ethernet, …).

Evolution of the Technology:
In the past POS (IP over SONET/SDH) was used, now we use MPLS over Ethernet.

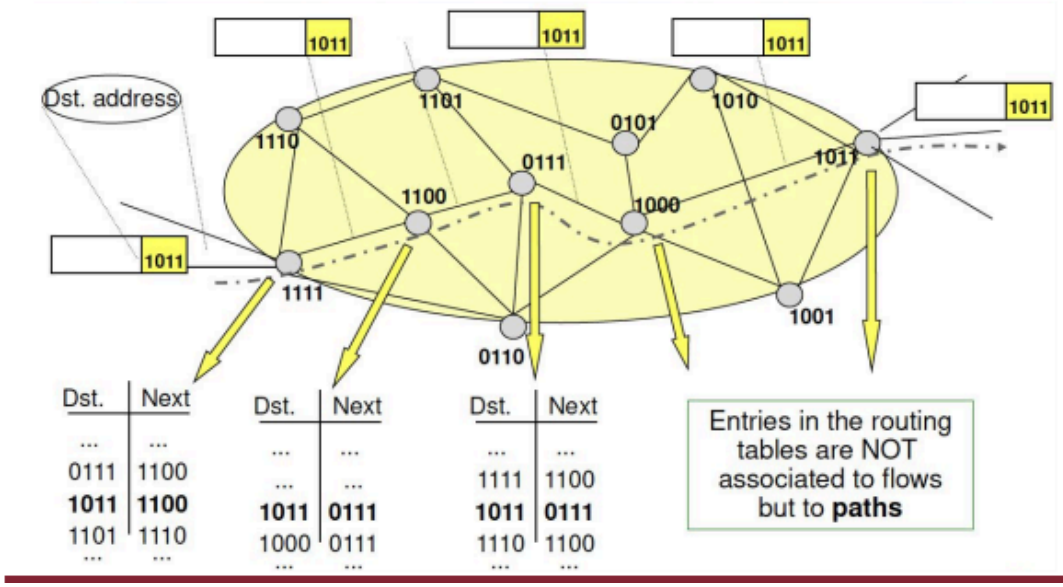


As shown in the picture, below the IP layer we have a 2.5 layer, that is the MPLS layer, that will support the three above-mentioned functionalities. Next step in the future will be to replace MPLS with SRv6 (segment routing).

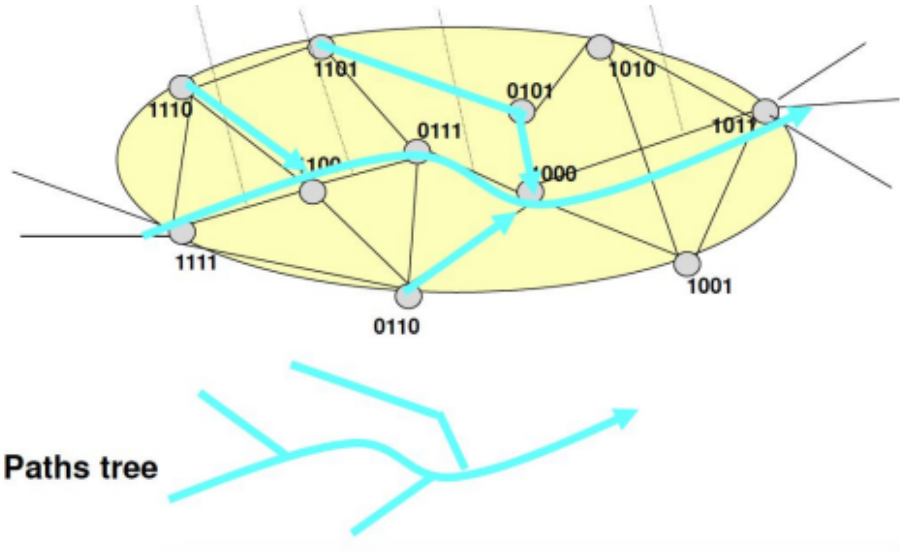


Connection-Oriented IP Network (MPLS):
IP is, by default, connectionless. In a connectionless environment dest and next in the *Traditional Routing Table* are IP addresses:

Connection-less packet network

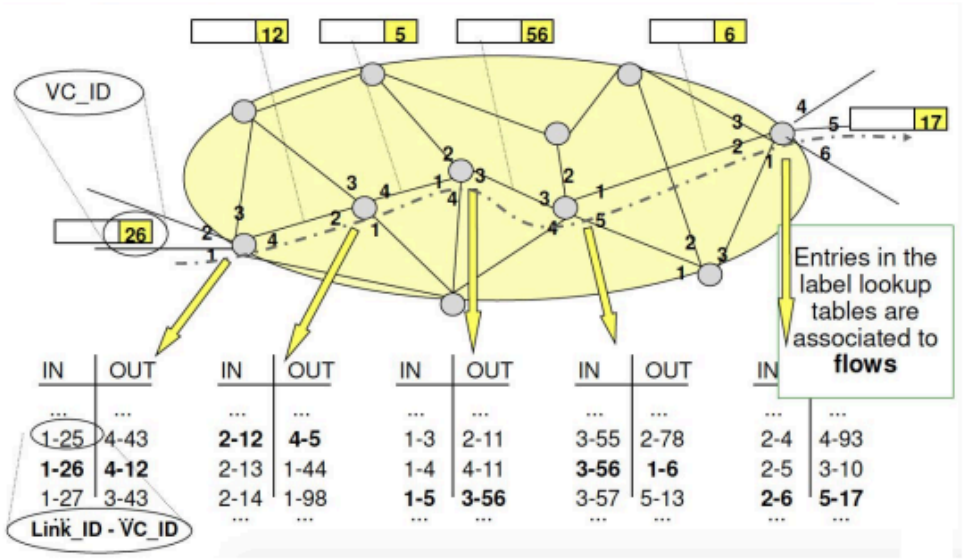


Due to Dijkstra, a path tree that has the destination as root is created and there is path aggregation, with the traffic converging in a single path.

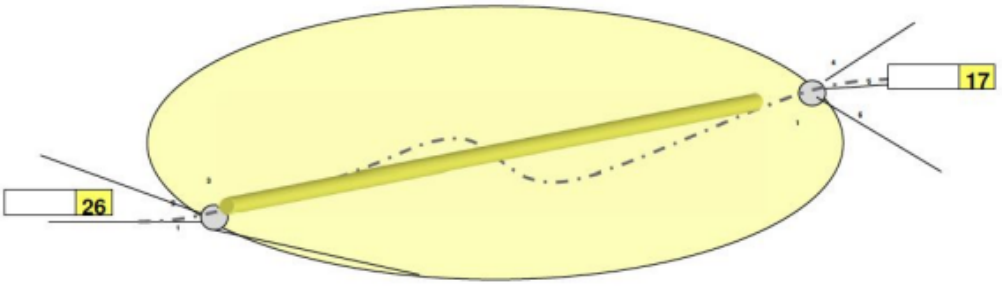


This definitely can't support traffic engineering or VPN, since packets always follow the shortest path.

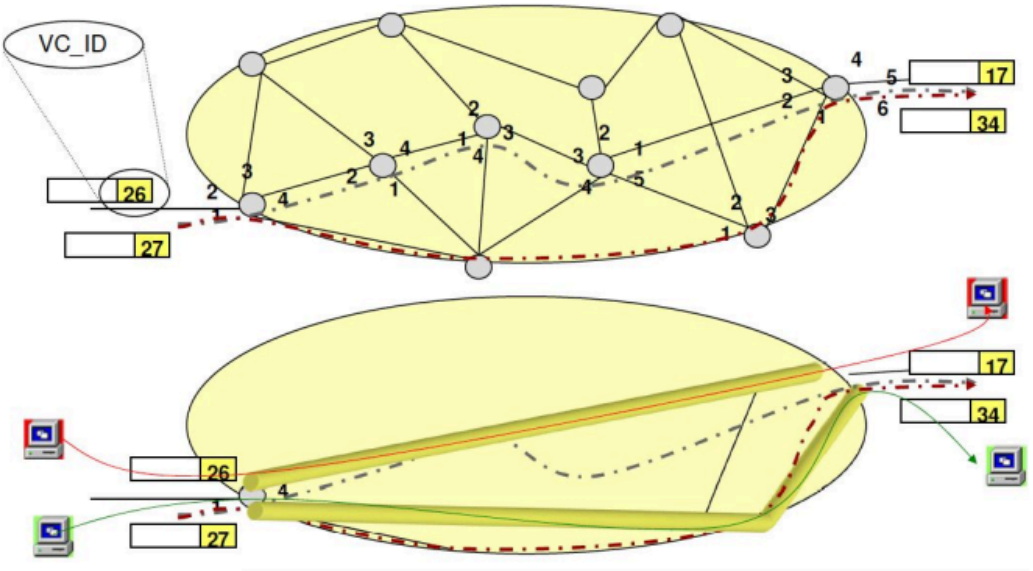
In an MPLS connection-oriented environment, we have this (assuming the connection is already up):



So the routing tables becomes a *Label Lookup Table*



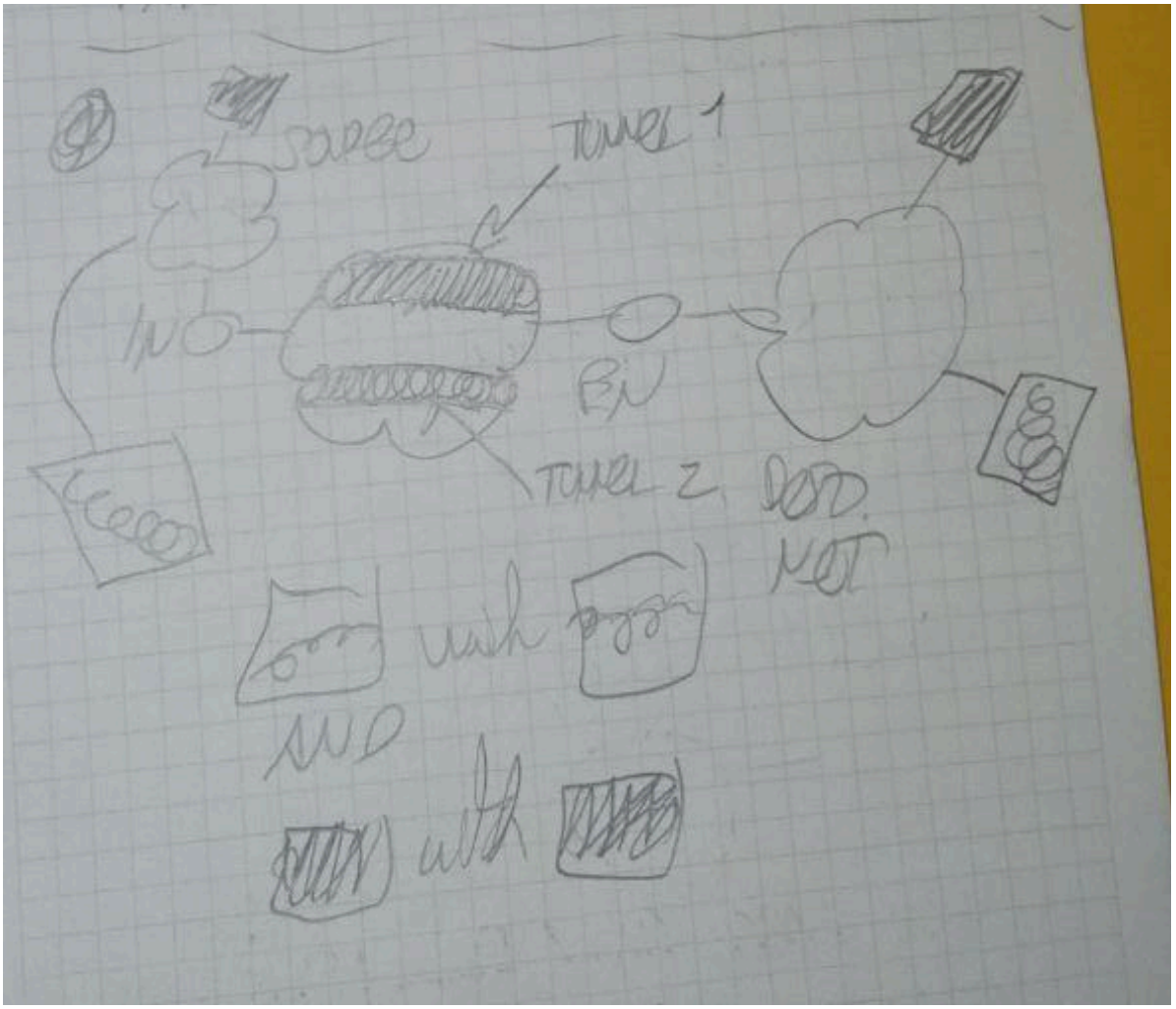
Now we have a tunnel. if we want the packet to follow the specific tunnel we just need to specify label 26. This way we can differentiate flows based on the label:



Let’s make an example: we have two different nodes in the source cloud that are sending their packets to IN. We want the packets to send 1 to tunnel 1 and 2 to tunnel 2. If “IN” is just an IP router there is no way to distinguish between the two, we want “IN” to perform the function classification. To classify means to distinguish between application flows. An application flow is composed by 5 fields that we have to inspect:

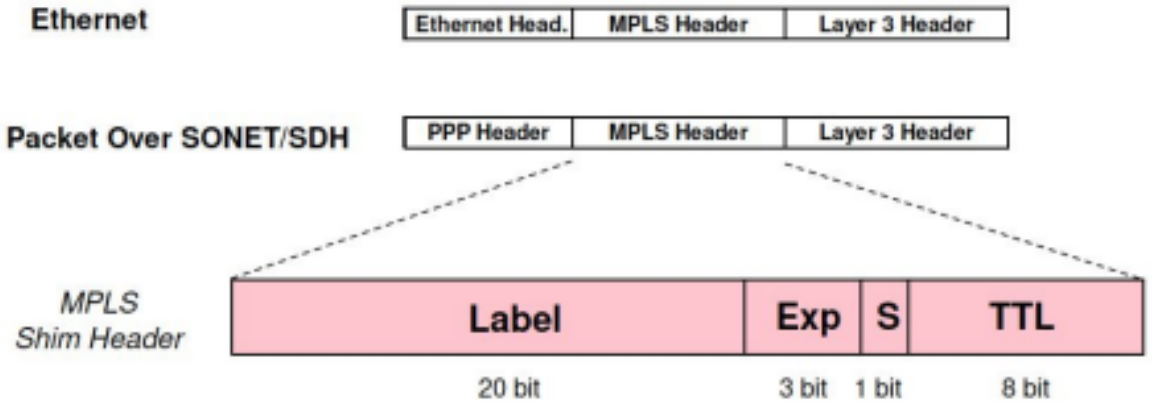
1. source IP
2. destination IP
3. protocol
4. source port
5. destination port

This is called 5-tuple classification. “IN” is the classifier; it will encapsulate the packet adding the label in the header, so that the net will route them correctly in one of the two tunnels.

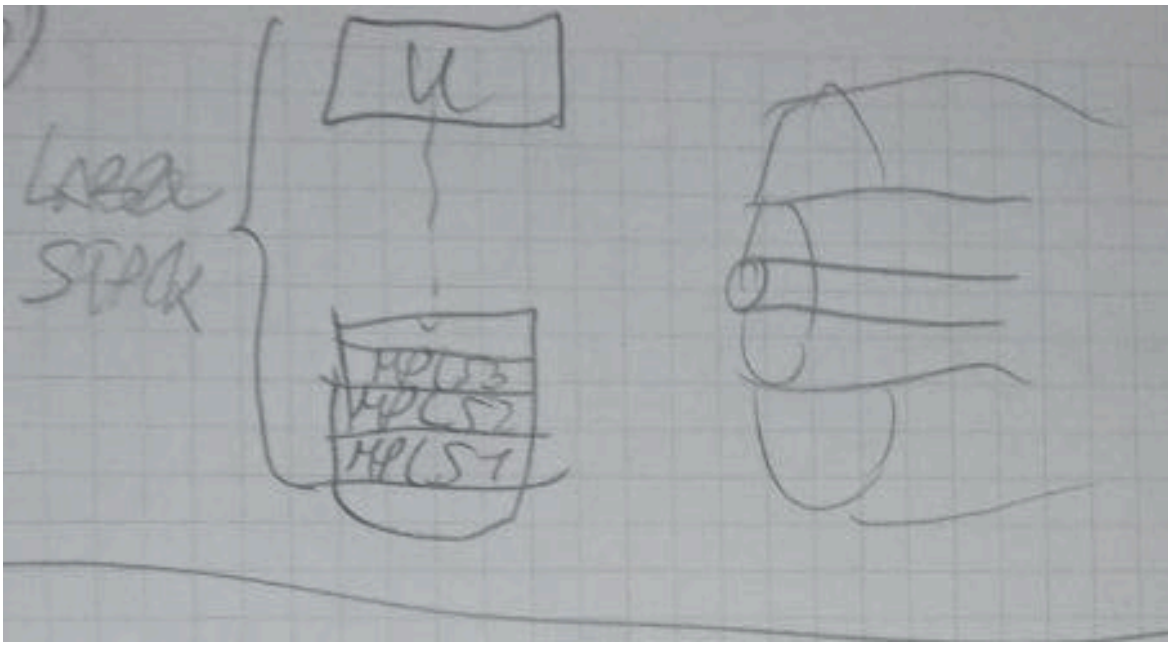


Note: what is the problem of loading 50% of traffic on tunnel 1 and the other 50% on tunnel 2? Out of order packet: imagine having packets from the same node in different tunnels; since each tunnel has its own delay, the packet will arrive at different times and maybe not in order.

MPLS Header:



MPLS header is inserted between L2 and L3 header. Its most important part is the Label. Then we have S, if S = 1 is the last MPLS header of the packet, after that this is the payload and header are finished. This means that we can have multiple MPLS header, in fact:



As shown in the picture, multilevel encapsulation means multiple tunnels one nested inside the other. Multi tunnels are useful for fast restoration and VPN.

MPLS Architecture (Terminology):

- LER are border nodes for an MPLS domain. It classifies IP packets (if they are entering) and forwards them to and from the MPLS domain, inserting and removing the label.

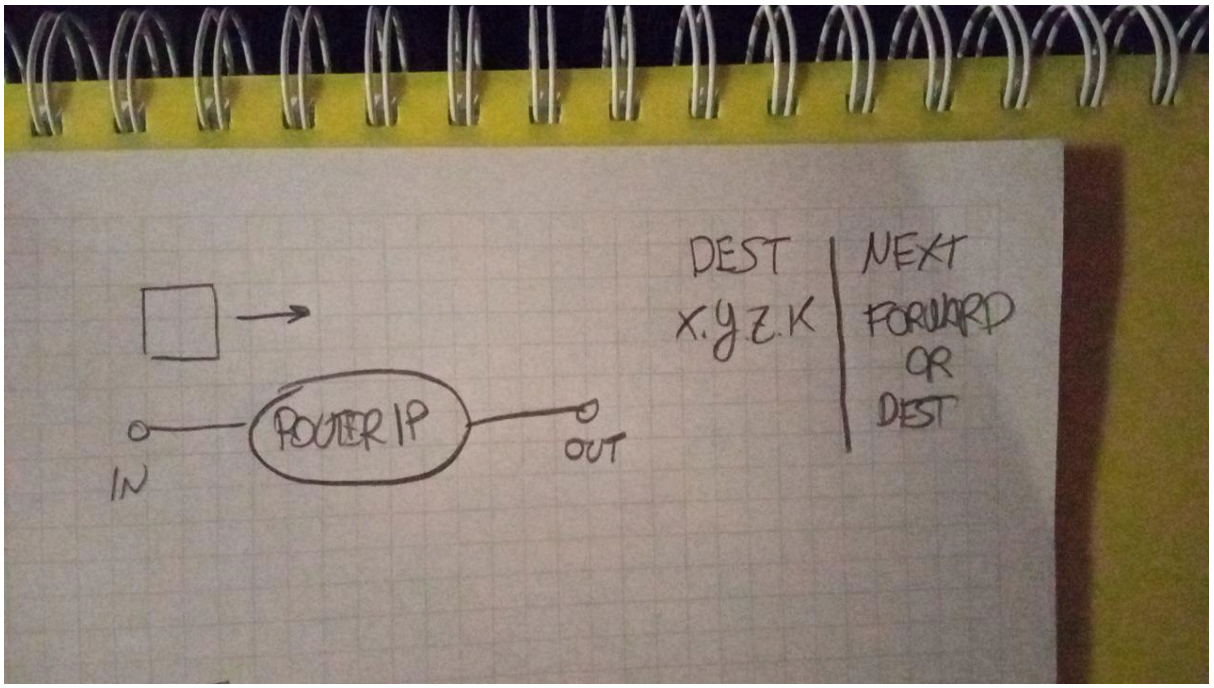
- LSR are routers that can do the label switching**.** They have
    - Control Component: level 3 intelligence (IP addressing and routing) + modules for label allocation, binding and exchanging label information with neighbors
    - Forwarding component: L2 switch but with label

- LDP: the protocol used together with traditional IP routing protocols to distribute the Labels among MPLS devices.

- FEC: a set of packets belonging to the same traffic flow

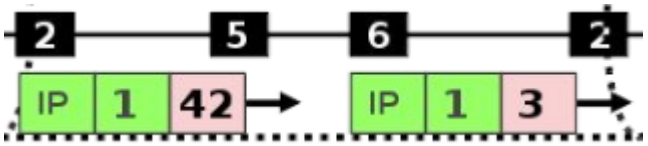
- LSP: we refer to an MPLS tunnel with this name. It's a virtual circuit. Packets from the same FEC follow the same LSP.

MPLS Functions (POP, PUSH, SWAP)
In traditional IP Routers:



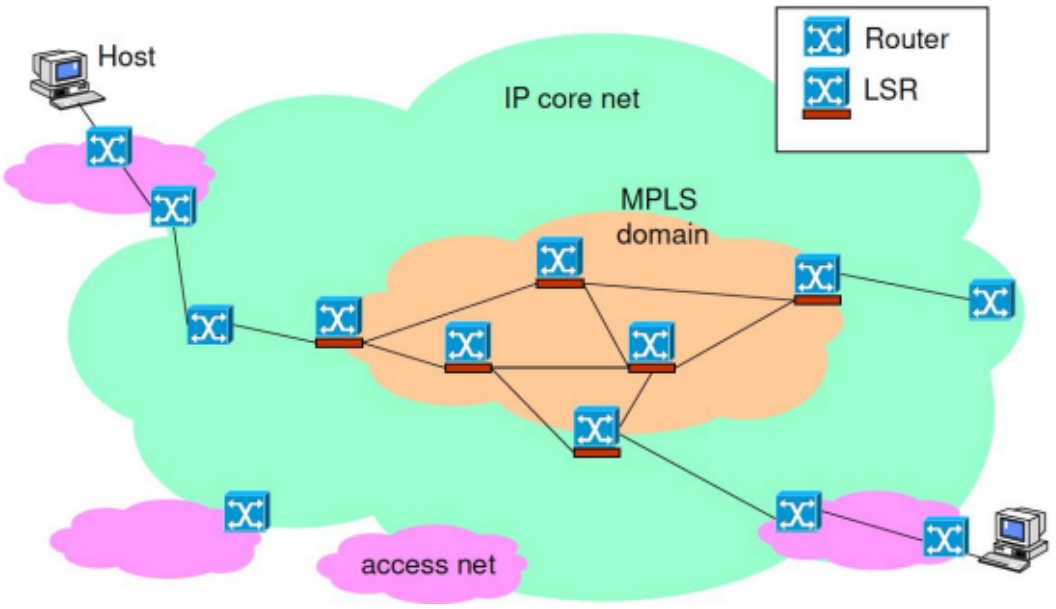
This is a *Match-Action paradigm*, in which we use longest prefix matching

In LSR instead, we match labels, so there is exact matching and, besides FORWARD and DROP, present also in IP Routers, we have SWAP labels, PUSH and POP MPLS headers functions. PUSH is needed to enter the tunnel and is done by an edged device, POP to exit the tunnel and go to the IP domain, and so it's done at the egress point. SWAP changes the ingress label with an egress label, e.g.

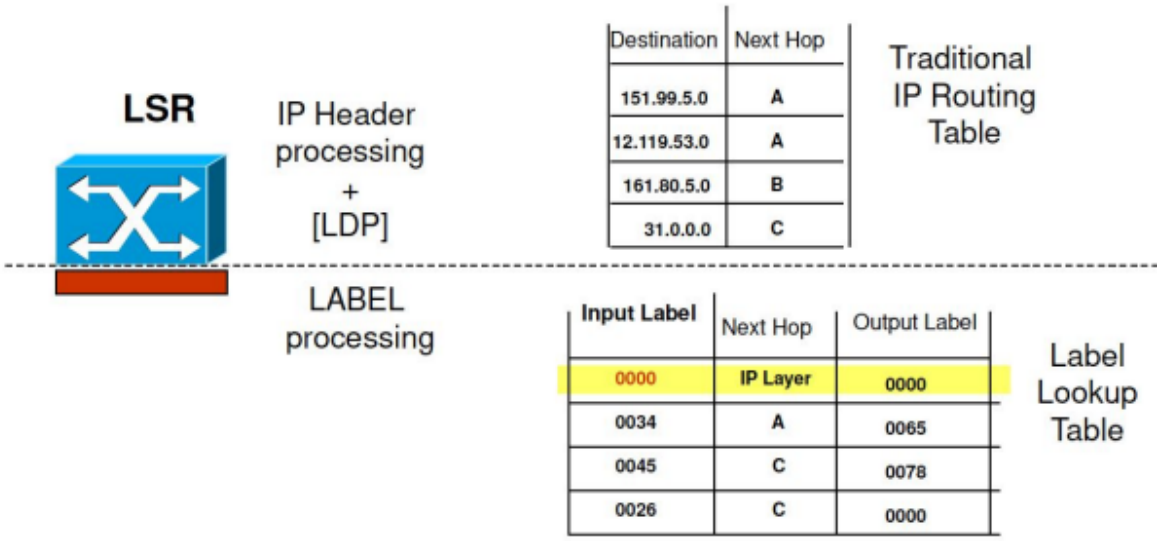


42 swapped with 3.

MPLS Domain within an IP Core Network



As shown in the picture below, when we have to exit the MPLS tunnel, and we start to rely only on IP, we just need to insert label 0000.



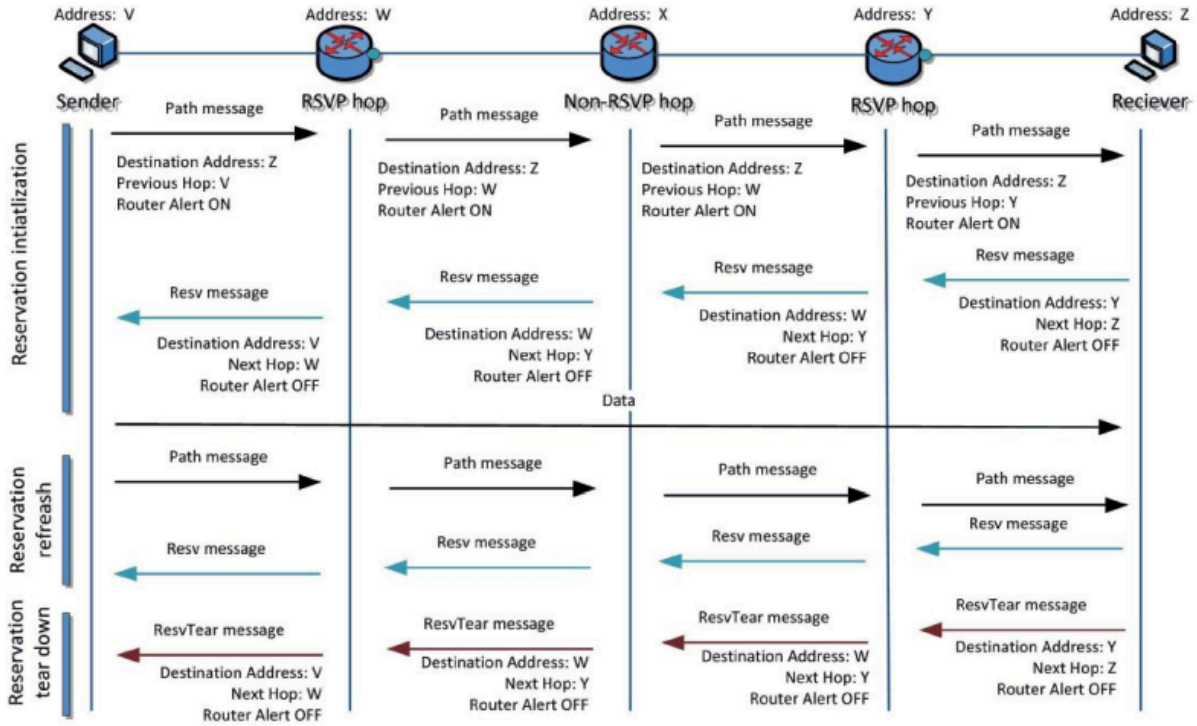
LSP Setup with LDP

So, when we want to create an LSP, these are the steps:

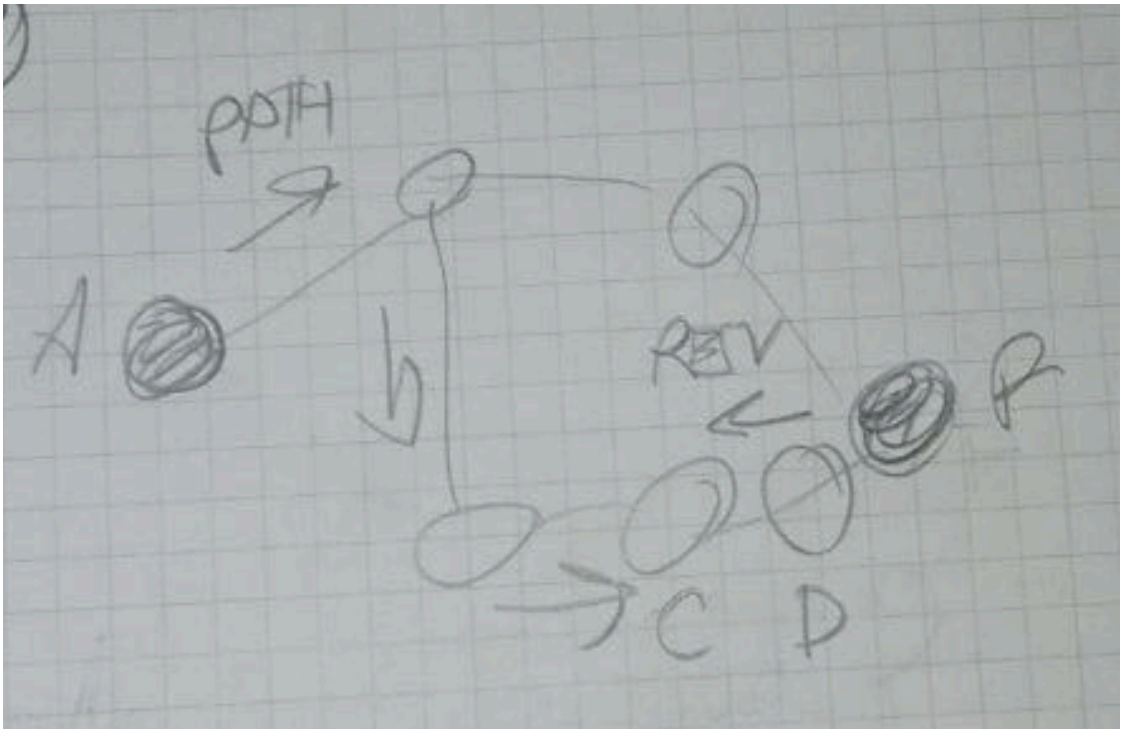
1. Request from the client
2. Net looks for a path with given bandwidth etc. (not necessarily the shortest)
3. instruct the switches to how to deal with the packets

For doing point 3, we can use a simple Label Distribution Protocol:

a. The sender sends a path message that, following the path chosen by the network, requests to allocate some labels on the different devices to create the LSP.
b. The receiver sends a Resv message back that specifies the label it expects

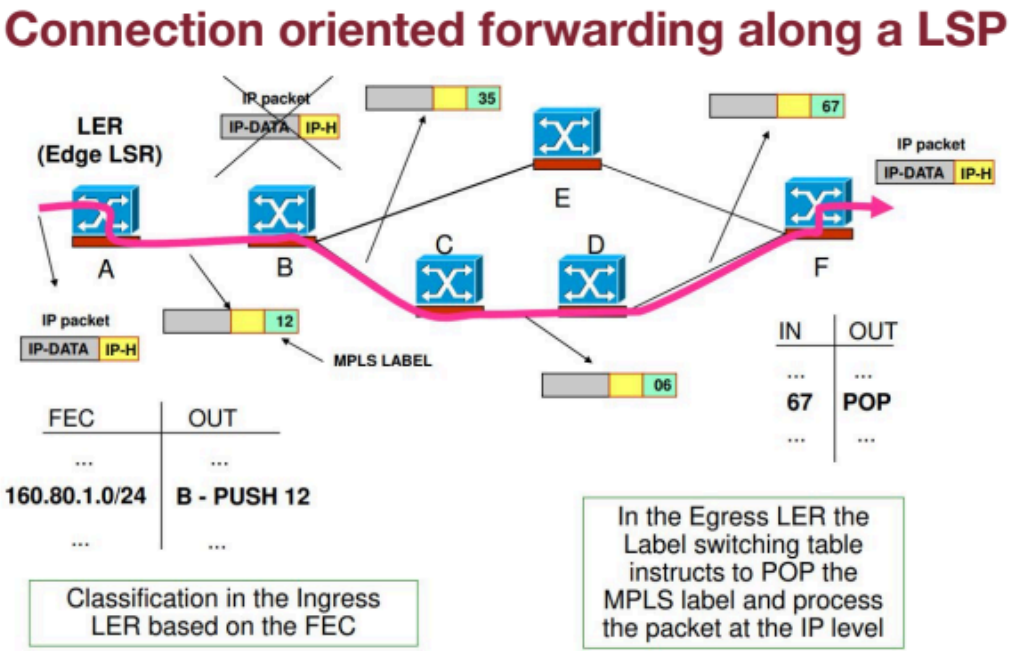


Let's make an example:



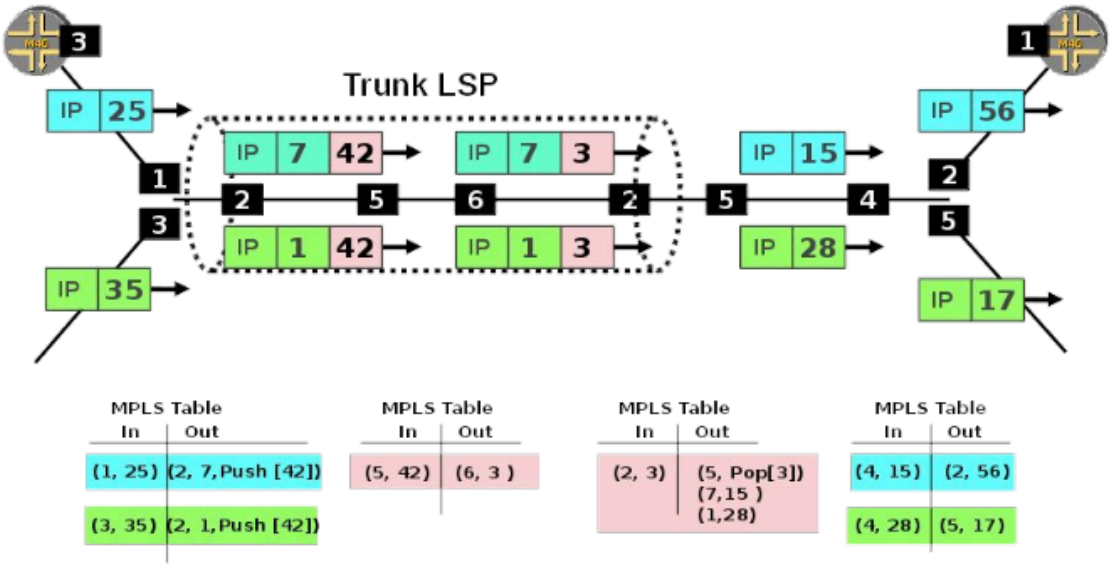
a. A sends the Path message that eventually reaches F.
b. F puts in his Label Switching Table 67 | POP 67 and sends back a RESV message F - SWAP 67 to D, with label 67 generated randomly.
c. D receives the RESV F - SWAP 67, selects 45 randomly, and put 45 | F - SWAP 67 and sends to D - SWAP 45 to C.
d. C goes 70 | D - SWAP 45

e. …
f. B, receiving C - SWAP x from C, chooses 12 randomly and send B - PUSH 12 to A
g. A, *that is the LER*, knows the application flows (5-tuple FEC) and has to classify them. So it insert in its Label Switching Table FEC | B - PUSH 12.

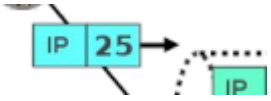


Note: here FEC is simplified, has only 1 field whereas it should be have all 5 field

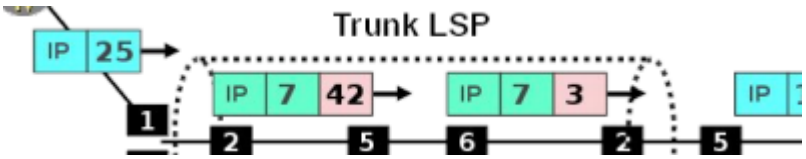
Label Stack (Trunk):



An LSP is a tunnel, while a Trunk LSP is a set of nested tunnels. We can do that by allowing internal nodes to do the PUSH operation.

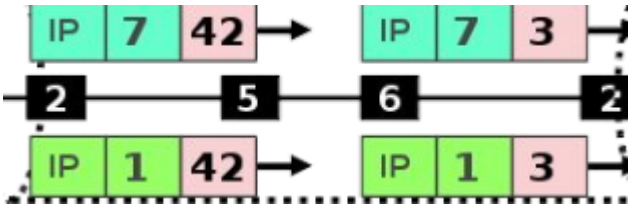


As we can see from the picture above, this packet is already traveling in a tunnel because it has a label



Now it has 2 labels.

The node 5 will read the outer label, just the 42, the 7 is hidden



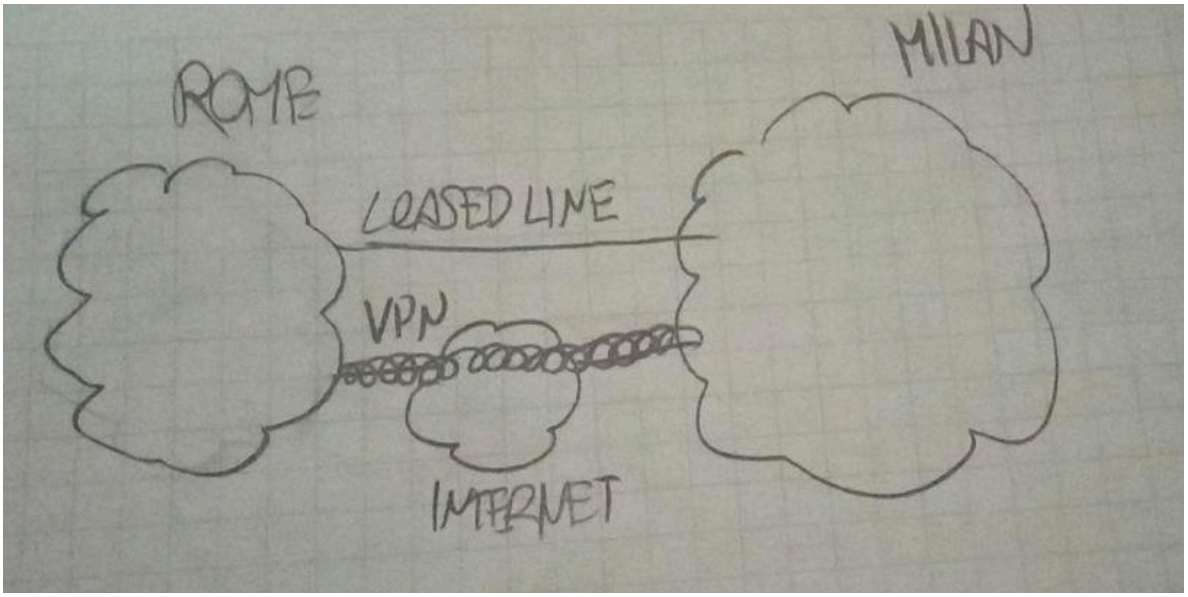
as you see the internal label doesn't change.

A possible question in the intermediate test can be: *how many nesting levels there are? Just count the number of labels.*

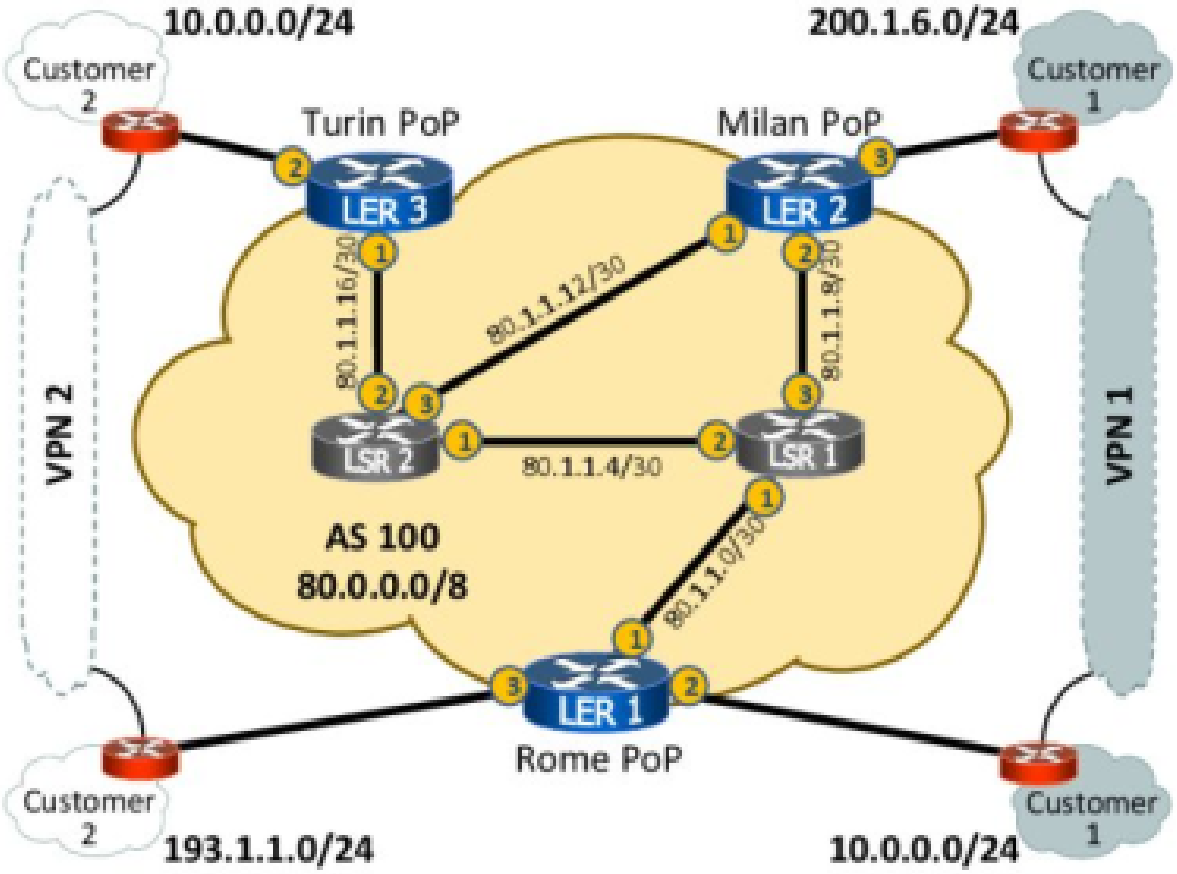
Now we will describe in depth the three services that a L3 provider should support.

VPN Service:
The main features of VPN are security/privacy and support of private IP addressing. We have two private networks one in Rome and the other in Milan and we want to connect them. We have two options:

a. Leased Line: an actual private line. This is unrealistic and expensive
b. We can use the public network (Internet), creating a tunnel with which only the two ends will communicate. For doing so we will need VPN (Virtual Private Network), since the two clouds use private address space that are not routable externally (in the Internet). A VPN encapsulates the packet in another packet: the outer packet will have a public IP that is routable on the Internet.

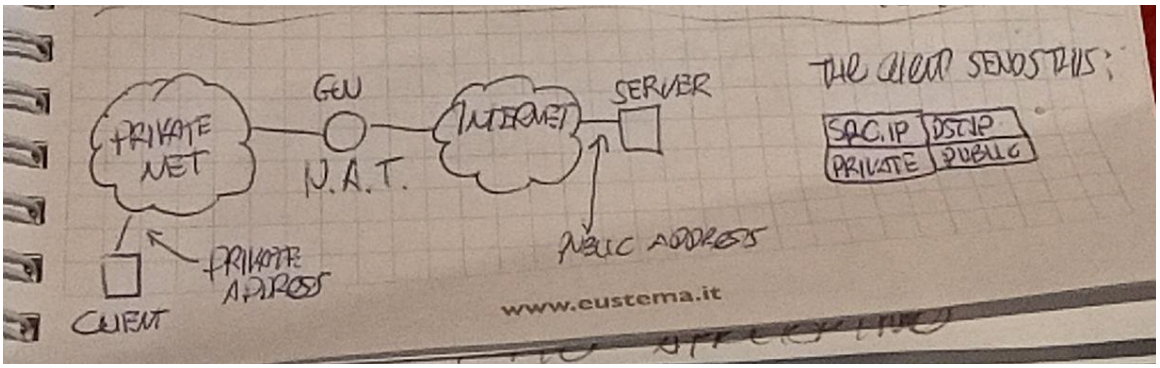


Let's consider, as an example, this MPLS network:

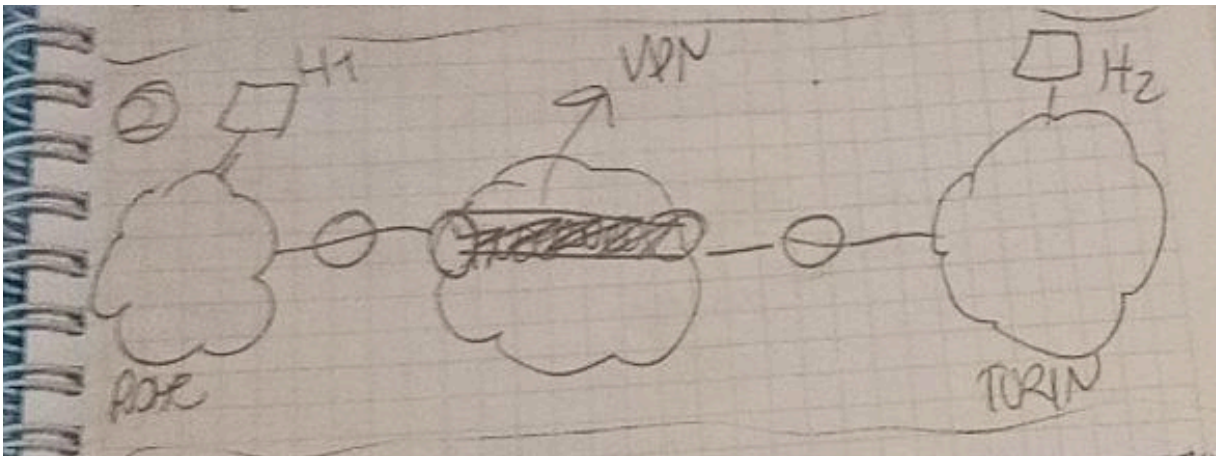


This is a TN that covers a wide area. There are two customers (e.g. 2 banks). Customer 2 has 2 networks: its employees in Turin would like to talk to the ones in Rome.

Can we use a NAT protocol on a GW for doing so? Let's see:



No, we can't because in our case both networks are private, so either src.ip and dst.ip are private addresses. We need a VPN

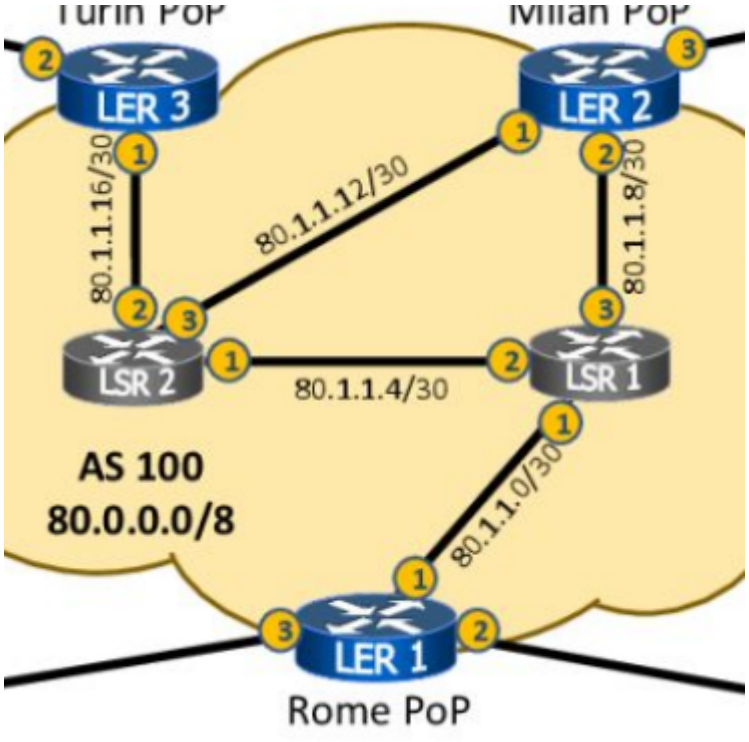


A VPN is a tunnel realized over a public network, in which we send packets. The service provider has multiple customers, and each wants to realize its VPN.

All the devices belonging to the MPLS net are called PE (Provider Edge), as this one:

Whereas the devices owned by customer are CE (Customer Edge), like this one:

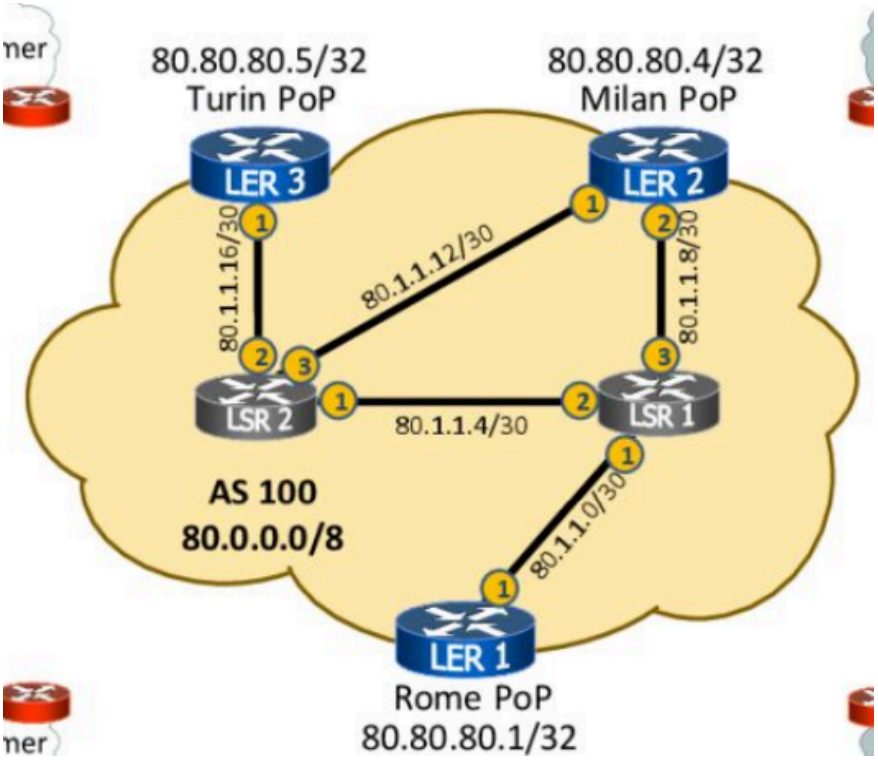
CE belongs to the customer so it's their business to manage them, whereas all the complexity of the VPN must stay in PE, so it must stay here:



the CE doesn't have to do anything for the VPN, just plug the cable.

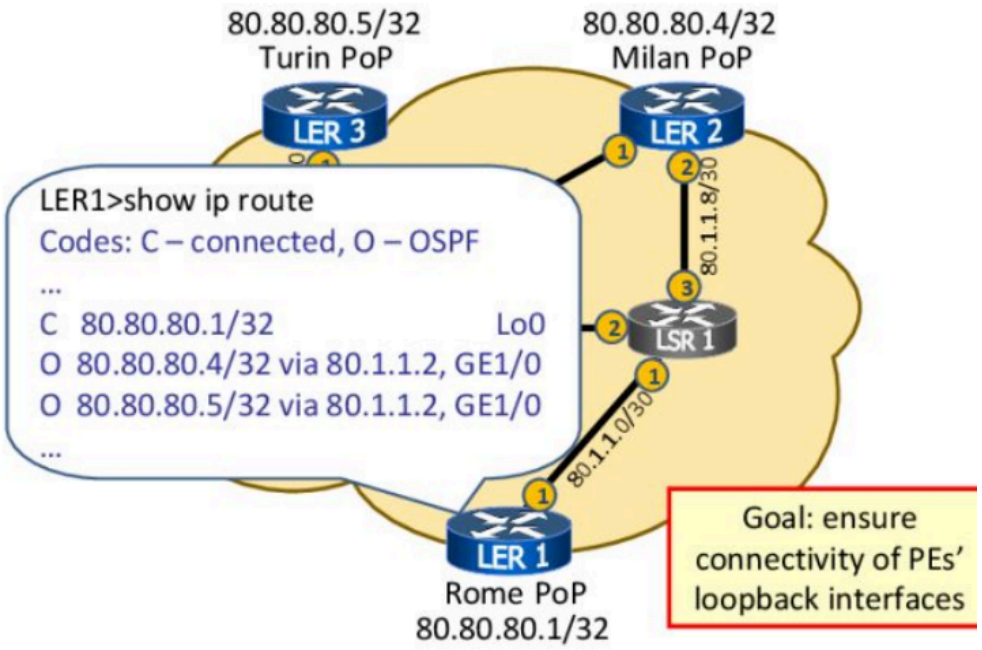
That's how you do that:

1. Move 1: achieve any-to-any IP connectivity among PEs



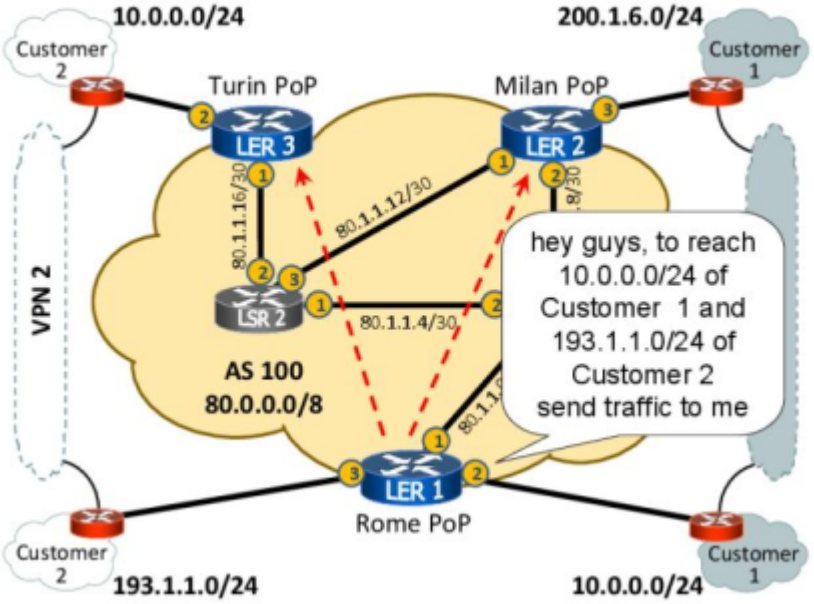
So we want to assign an ID to every single PE (at the border!). We use a loopback interface and assign to it an IP. Why do we identify the router like that, with a loopback interface? The reason is that physical interfaces can be up or down, due to failures or other reasons. On the other hand, logical interfaces such as loopback interfaces are always up, they are stable over time. So we use it as an ID.

We must be sure then that every router knows how to reach the others. For doing so we use a routing algorithm that lets reachability information travel: the OSPF protocol.



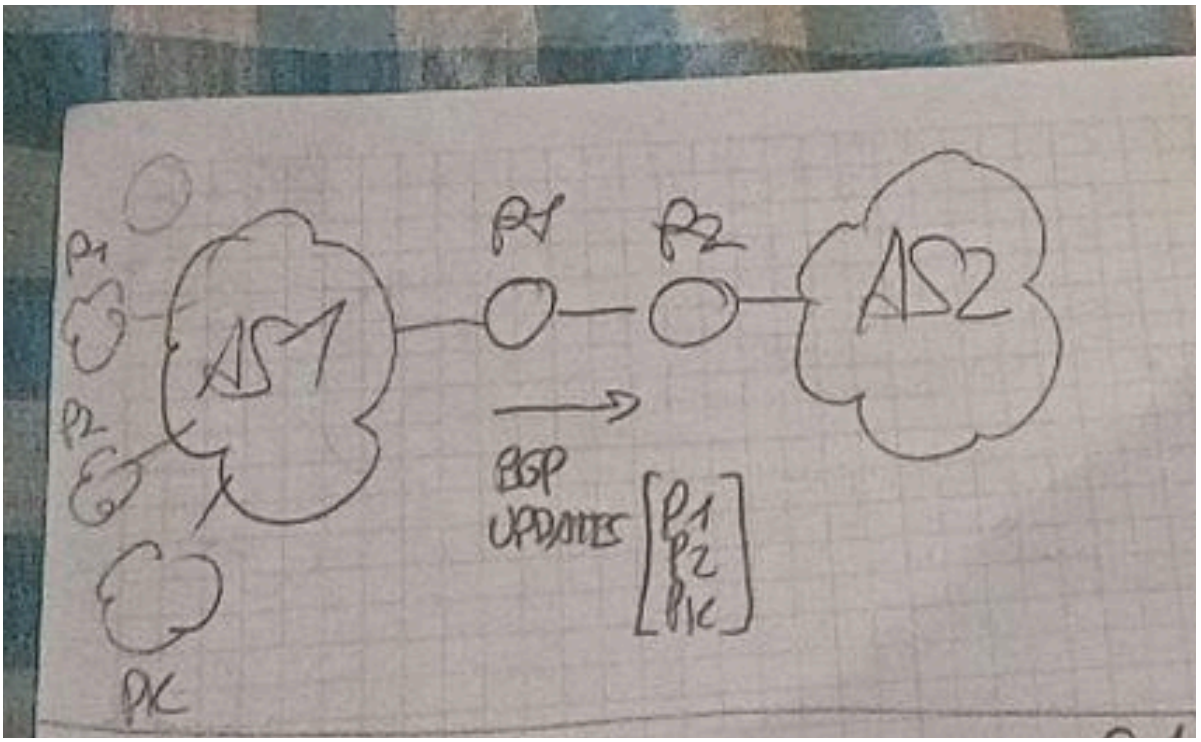
As shown in the picture, the Rome node knows how to reach the other router of the PE.

2. Move 2: define a signaling mechanism to distribute customer prefixes among PEs



Rome PoP is giving access to 2 private nets, customer 1 and 2 nets. How can Turin Pop and Milan PoP be aware of the existence of these two CE?

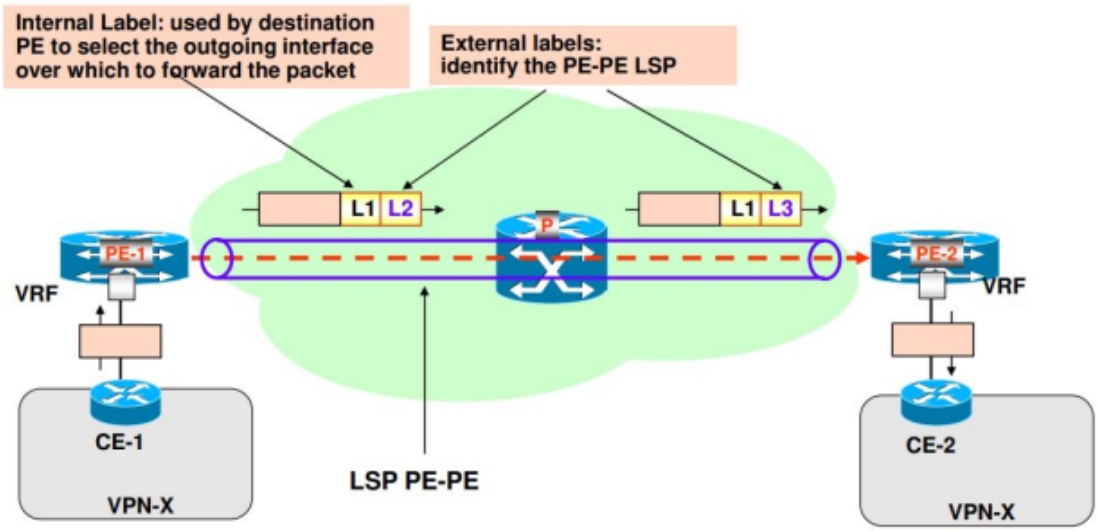
we use BGP: allows two nodes to exchange messages that contain all the reachable prefixes. Example:



Note: AS stands for Autonomous system. After the BGP updates R2 knows that AS1 has p1, p2, …, pk prefixes and so it knows that these nodes are reachable through R1.

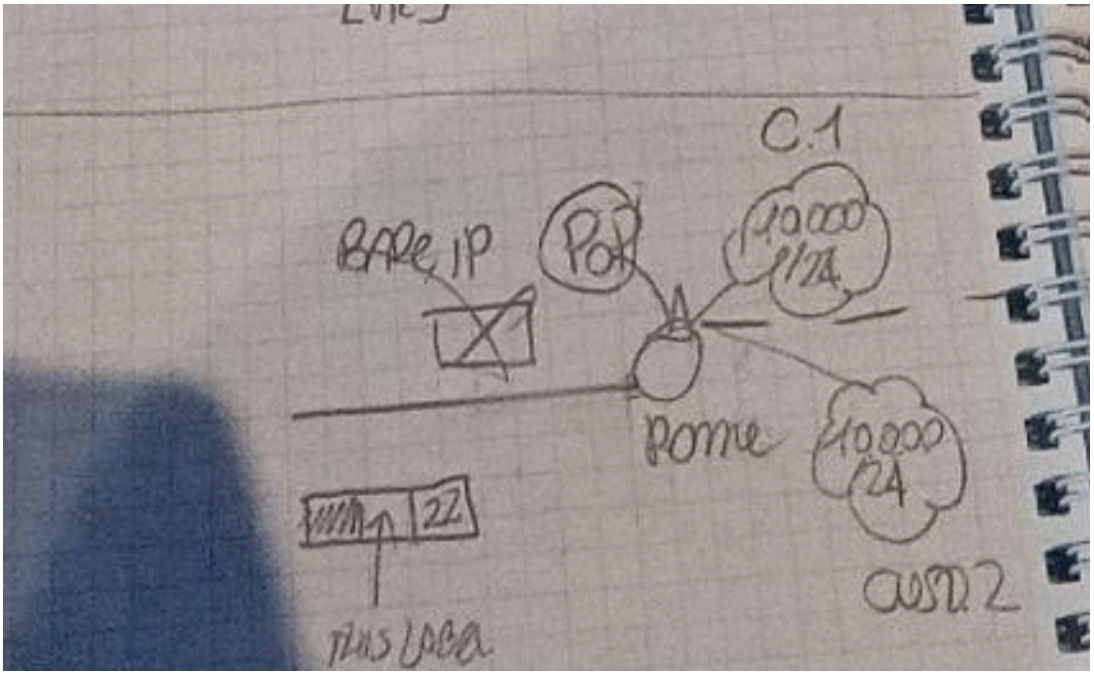
So we want to have BGP sessions between all the PEs in the provider network. But there is a problem: there is no problem in having 2 different routers that both say “hey through me you reach node x”. But here Rome PoP says “you can reach 10.0.0.0 through me” and idem Turin Pop, but they are two different private nets with the same IP 10.0.0.0. For solving this we use BGP modified: notify a list of entities (L3VPN identifier) not addresses, so there is no ambiguity.

3. Move 3: define an encapsulation mechanism to transport packets from one PE to another across the network



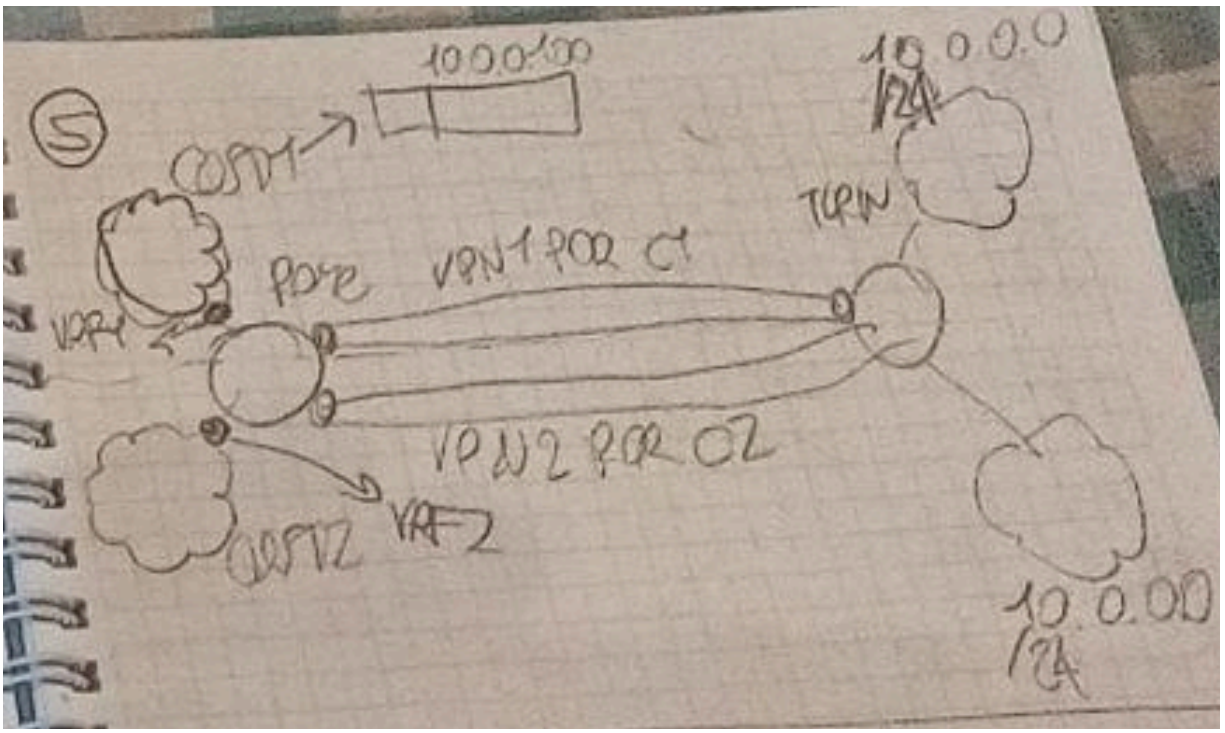
Until now, with the first 2 moves, the packet still carries private addresses. Now it's time to create a label switched path. If we use labels no matter what address is there, even private, the router will not read them; it will just read the labels.

There is a problem: let's assume we have this situation:

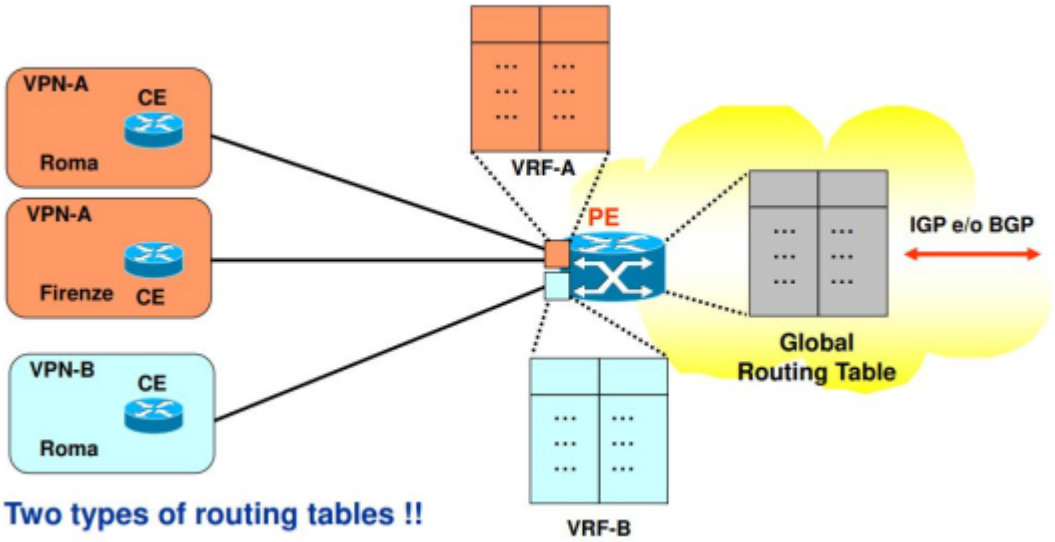


The node does the POP operation and it's left with a bare IP packet, so it has to use only the IP address to deliver it. The problem is, there is ambiguity. To solve that let's create a trunk, with the internal label that tells us which of the two it's our destination (the label addressed with "this label" in the picture)

There is a last problem to address. Let's consider this situation:



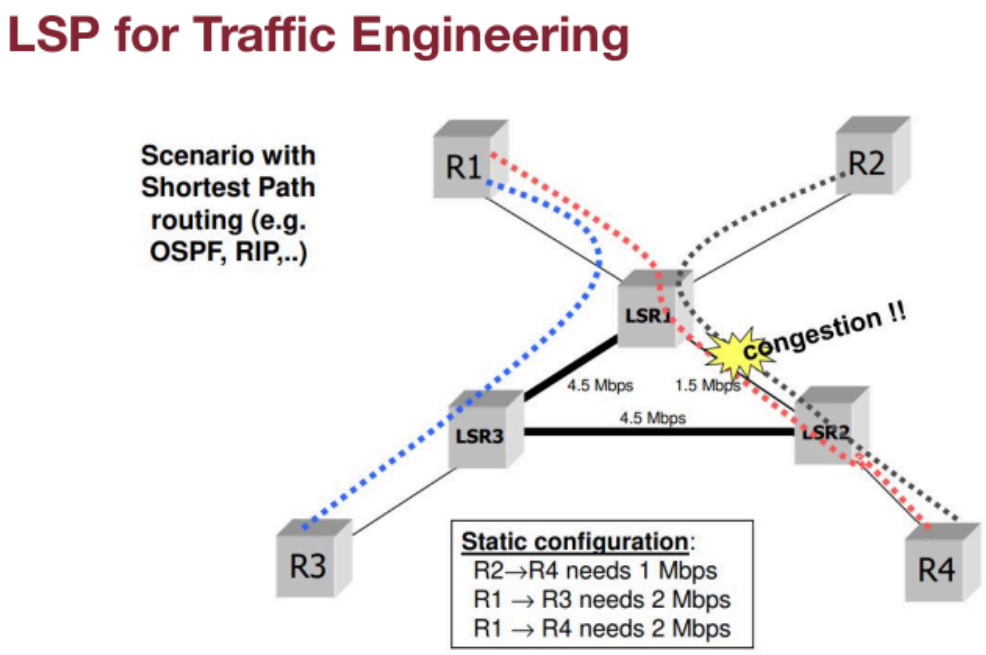
customer 1 sends a packet to destination 10.0.0.100, should it use VPN 1 or VPN 2? I have to define multiple routing tables, one for each customer. These routing tables are called VRF (Virtual Routing and Forwarding). When the packet comes from the interface connecting Customer 1 and Rome node, Rome node knows which VRF to inspect.



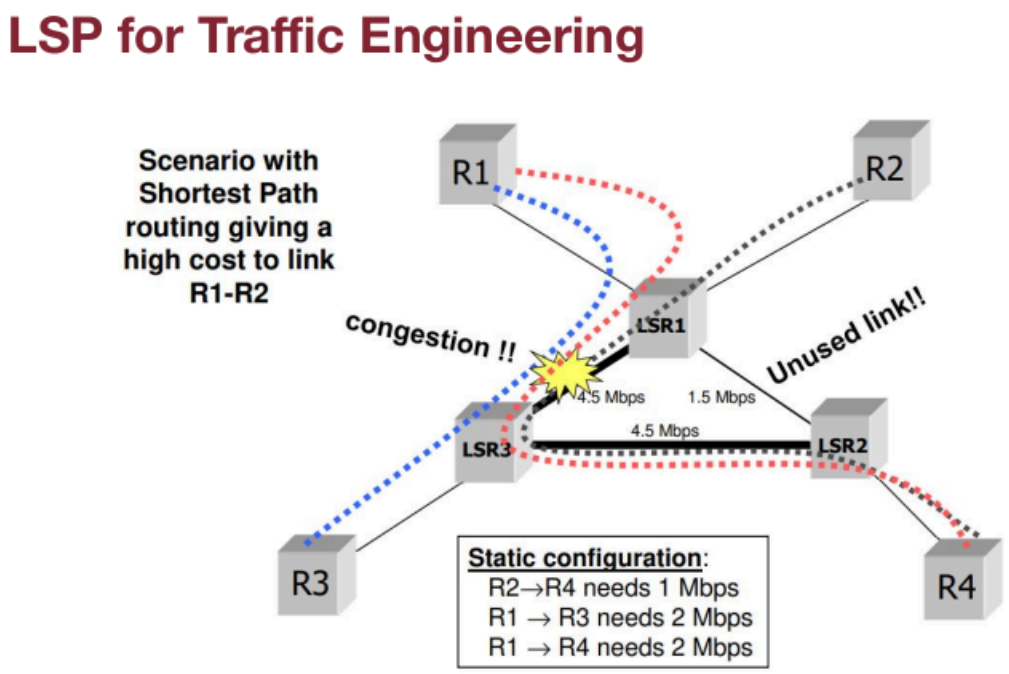
VRF are not necessarily one-to-one in terms of interfaces, but we have a VRF for each customer.

Traffic Engineering Service: in the context of routing, so finding a path from one point to another. For doing so in layer 3 we use routing tables, installing rules in it and we run

algorithms, such as Dijkstra. Dijkstra looks for the shortest path, but sometimes the shortest path is not the best choice. For instance:



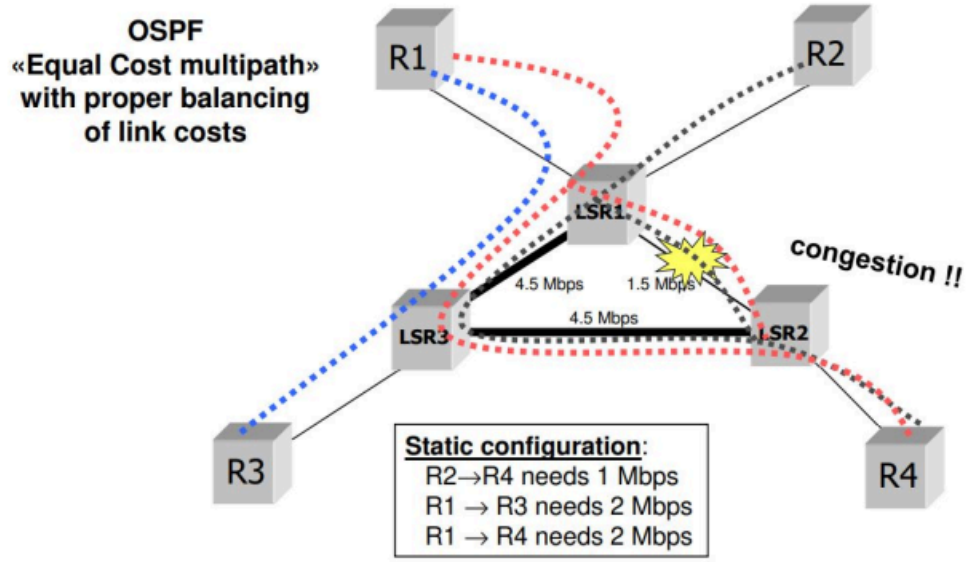
Dijkstra will solve the problem by just increasing the weight of passing through that link:



but this will make the net converge in another state, just moving the congestion in another place, as shown in the picture above.

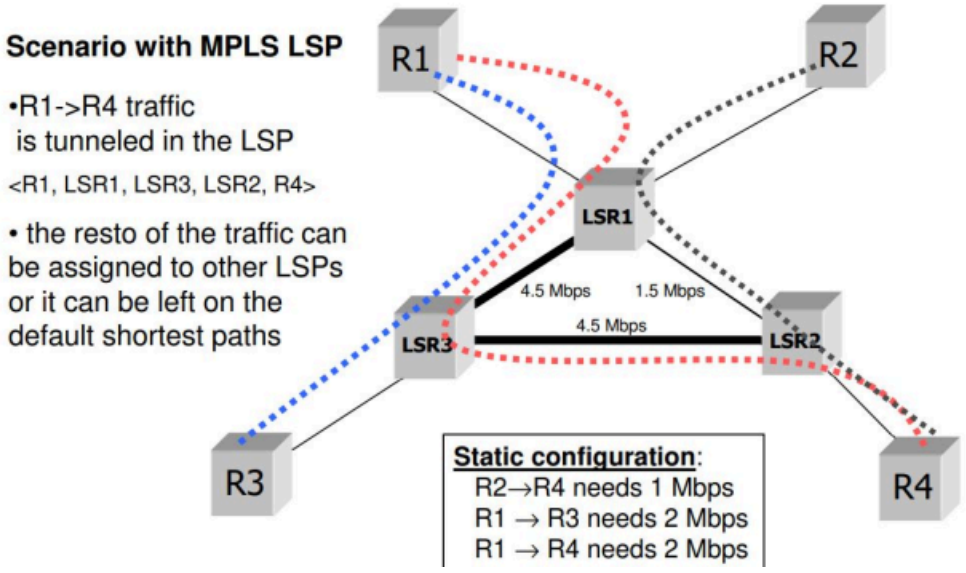
I could just split the traffic into the two path, but once more this create congestion:

LSP for Traffic Engineering



We would like to have a technology that supports traffic engineering: a function that allows a net operator to balance the load. MPLS can state, for instance, that red flow is the only one that can pass through this link. So we are actually creating a LSP.

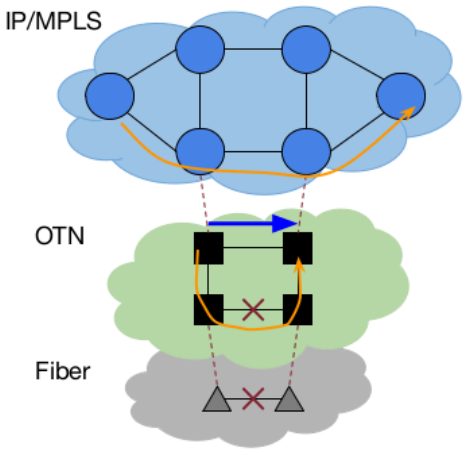
LSP for Traffic Engineering

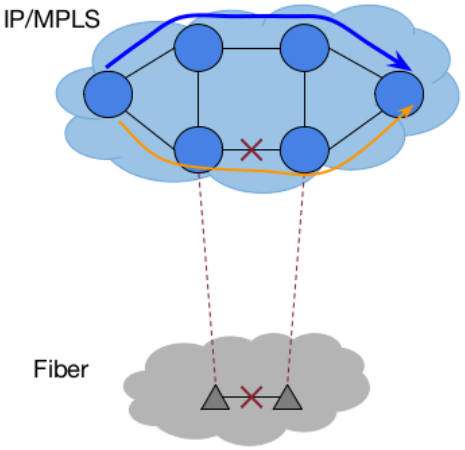


How can I find a router configuration like that? There are other uni subjects that explain this problem

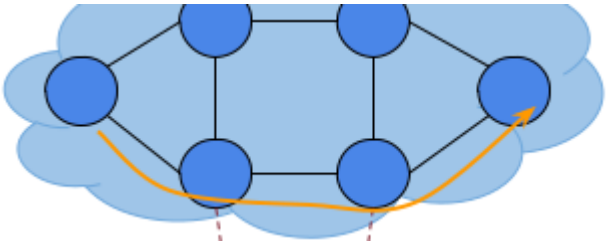
Fast Recovery after Failure Service: how does the IP net react to the failures? The net will find a new path but it takes time. So the provider can provide fast restoration, in the order of milliseconds, thanks to MPLS.

Let's consider this example:

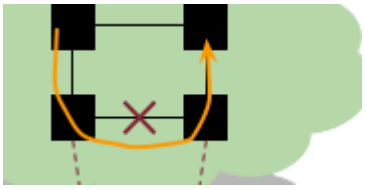




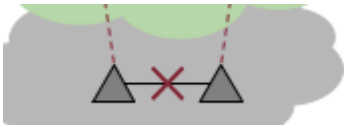
This IP/MPLS network:



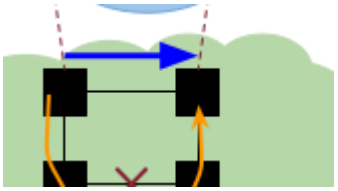
is created over this lightpath:



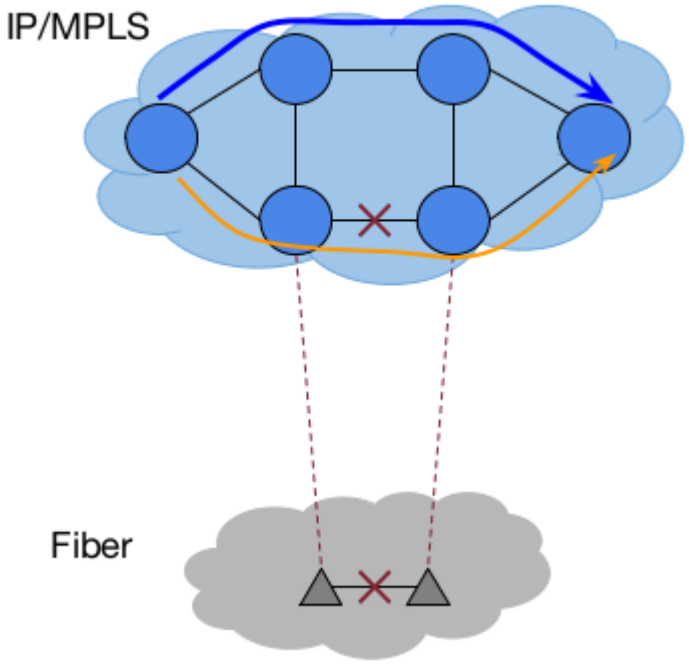
Let's assume this fiber fails:



After detecting the failure, I find that I can pass here (blue arrow)



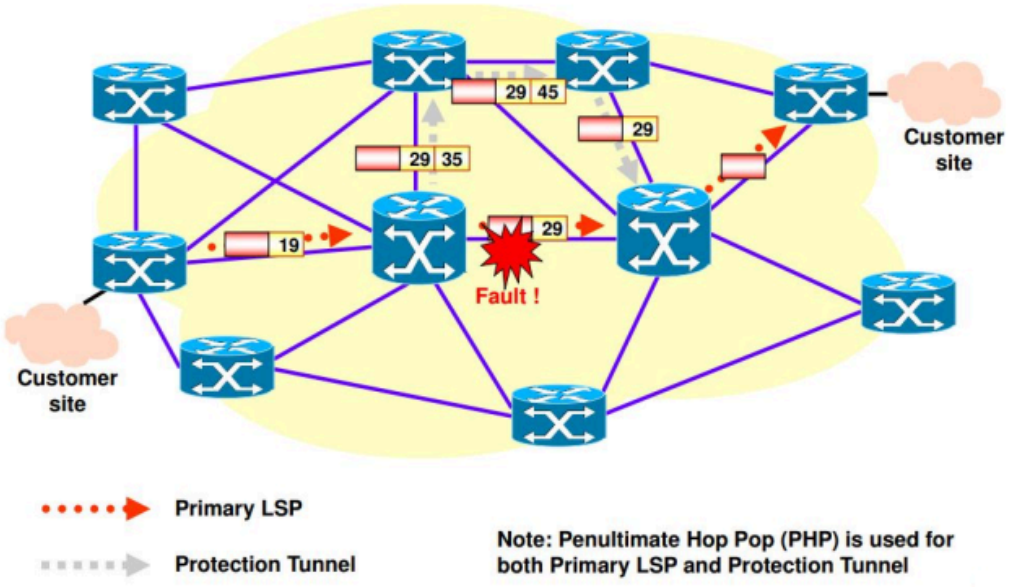
It would be nice to solve this problem on a higher level, the IP level.



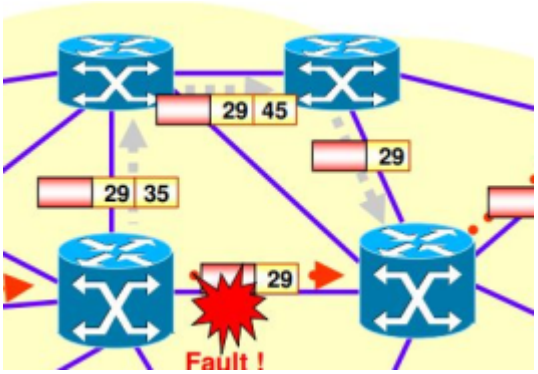
This way I don’t have to wait until the optical network has converged, because I can find the alternative path directly in the IP net.

How to implement this? We have two options:

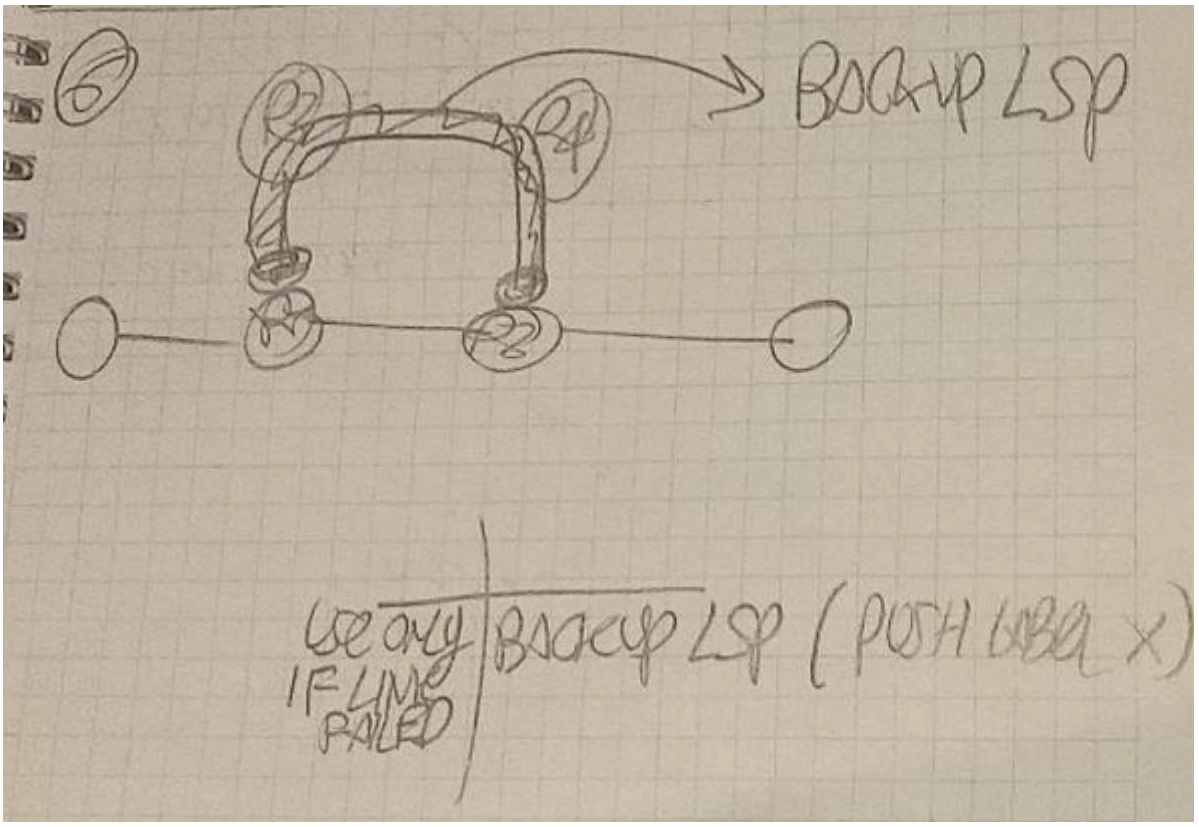
1. We can use bypass paths (FAST RE-ROUTING):



   for doing so we use LSP trunk:

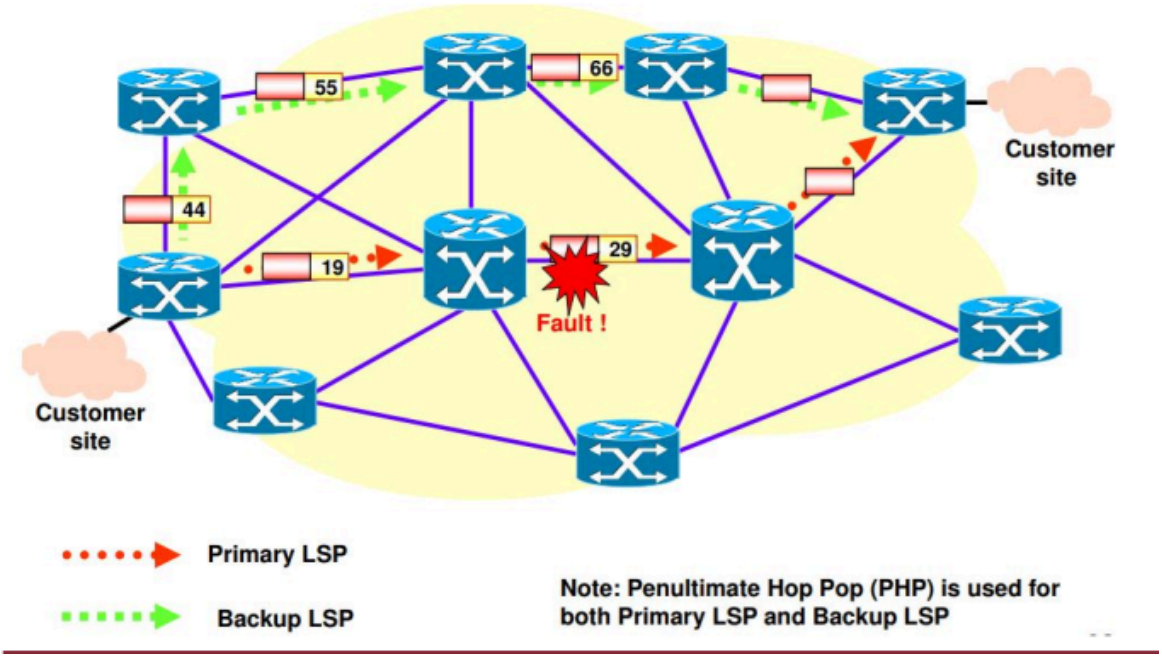


we have to push a label that make the packet follow the alternative path (gray arrow)



we can assume that in the R1 label switching table there is the command written in the image. It's a trunk because the incoming traffic is already labeled

2. instead of creating a bypass for every single link of the net, create a path ingress-egress (BACKUP LSP)



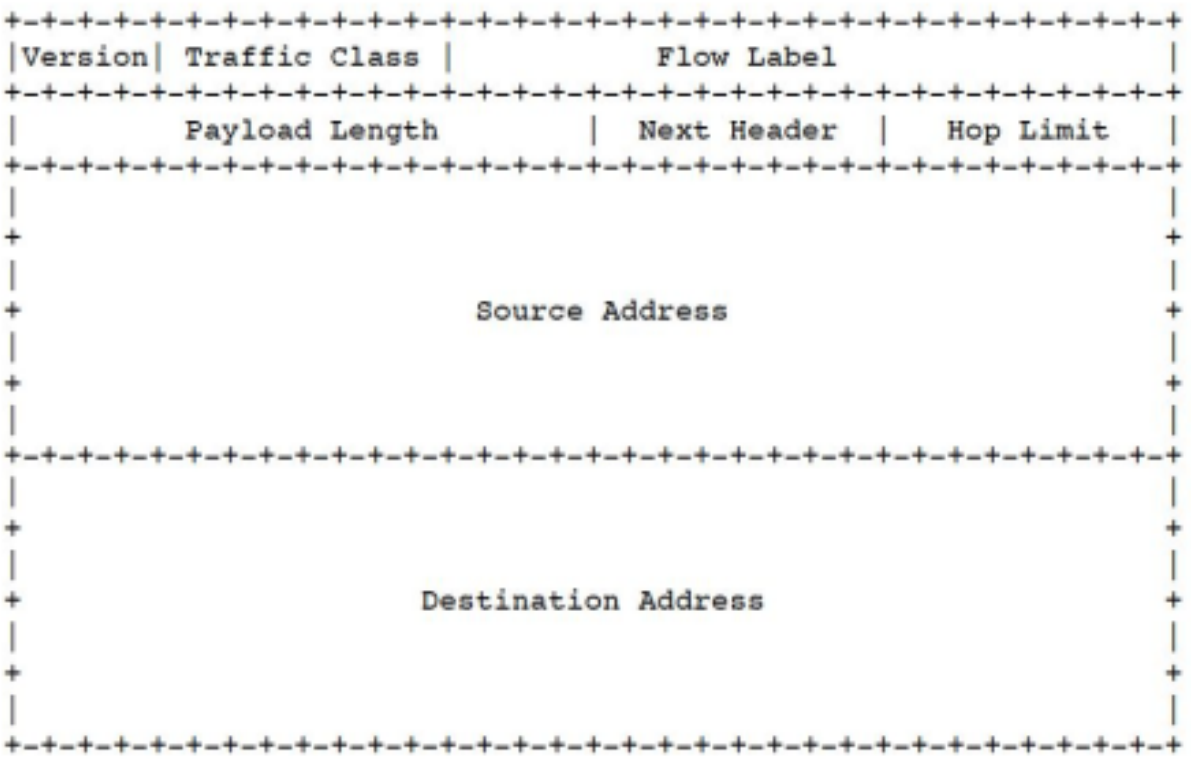
In this case the node has to be notified since the link is not contiguous but remote

Internet Protocol IPv6

16/10, 23/10 and 27/10

32 bit for ipv4 so we have 2^32 ipv4 addresses, while 128 bit for IPv6 for 2^128 addresses. We were running out of IPv4 addresses. NAT was the mid/short term solution for it, but today is still widely used. Ipv6 is the long term solution, with it we don’t have any problem of expiring the address space.

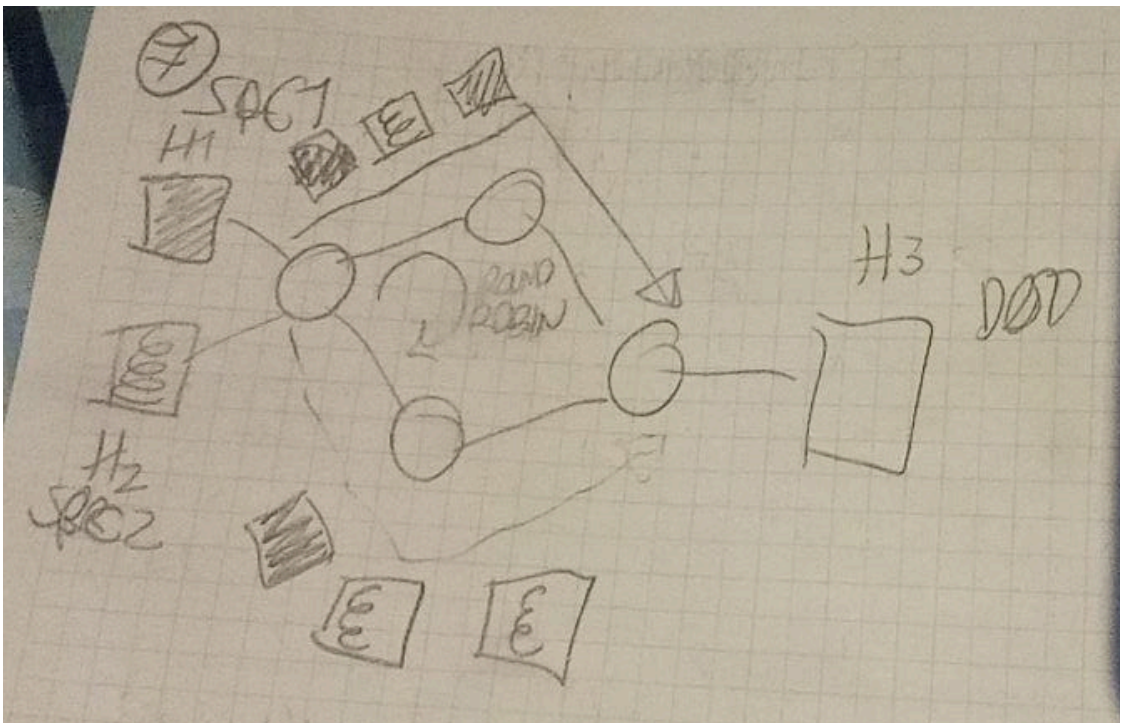
IPv6 Header:



IPv6 is an ***extensible*** protocol, so we don't have to design a new protocol each time we want to add a feature. That was not the case for IPv4: for instance, when IPv4 was created, security was not a concern. Now it is, and so the Ipsec protocol was developed. Moreover, IPv4 has a variable length header, and for a machine it is complex to handle it. In IPv6 we remove all the unnecessary things. For instance, fragmentation is not implemented by default, since if the packet is small there will not be fragments. If I need fragmentation, I just add an extension.

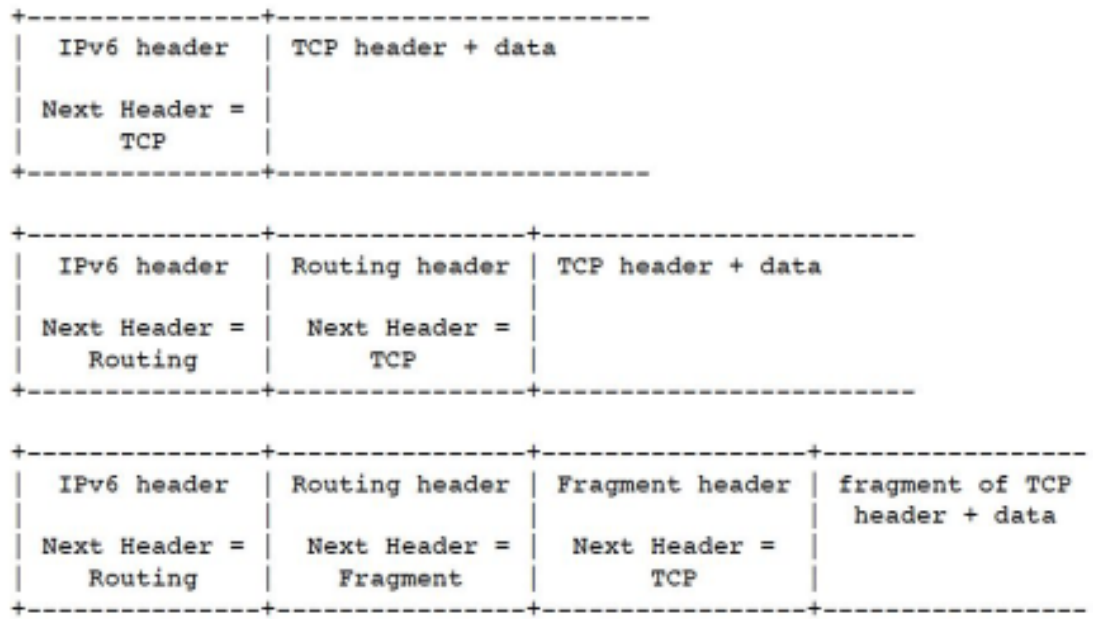
The only mandatory things we always have in the header are:

1. Src and Dst addresses: the majority of the space is due to the source and destination addresses, since we are using 128 bits
2. Version: IPv4 or IPv6
3. Traffic class: in IPv4 there is just best-effort. Here we can choose best-effort, or real time if we are dealing, for instance, with online gaming. So I can provide different levels of QoS
4. Flow Label. Let's consider this example:



we can either use the upper part or the down path since they have the same length (ECMP Equal Cost Multi Path). We can think of doing Round Robin, but there is a problem: layer 4 protocols, such as TCP, are sensitive to out-of-order packets. Acting this way TCP could decide to decrease the TH because the out-of-order window is expired. The solution is to put all H1 above and the H2 below, but how can the router distinguish the two flows? We have to write a Flow Label field in the header of Ipv6: some labels are marked “H1 flow” some others “H2 flow”.

5. Hop Limit: IPv4 TTL is Hop Limit in IPv6
6. Next header: specifies the kind of header that follows

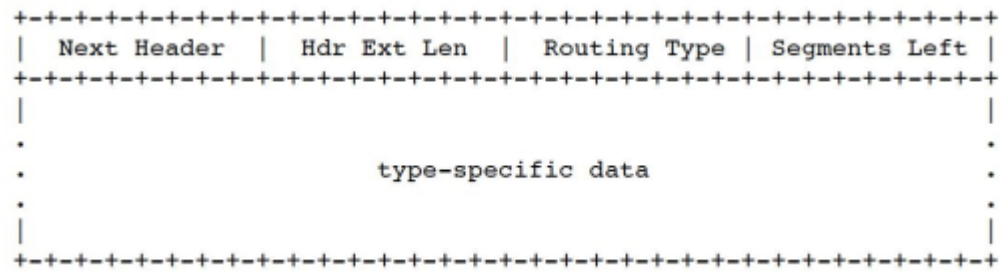


Types of headers:

1. IPv6 basic header
2. Hop-by-Hop Options. It must be processed in every node we encounter
3. Destination Options: header inspected only in the destination node
4. Routing header
5. Fragment Header
6. Authentication: if you want to authenticate a packet you have to add it
7. Encapsulating Security Payload: if you want to get confidentiality
8. Destination header
9. Upper-Layer header

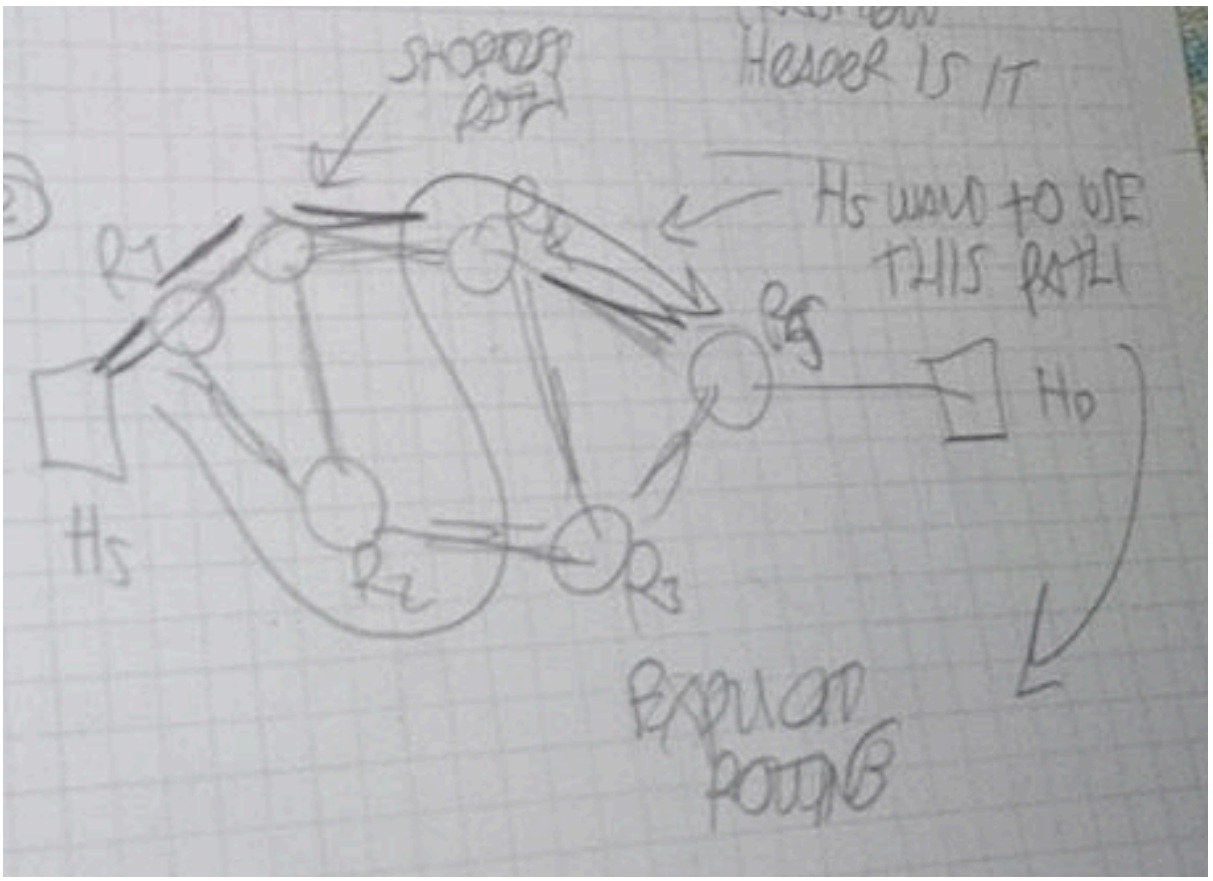
The basic header must always be the first in the line, and this is the recommended order

Routing header:
is used to specify the intermediate nodes we want to visit on the way to the destination.



Routing type specifies a particular variant of the header, Segment Left is the number of intermediate nodes in the list that still need to be visited.

Let’s make an example:

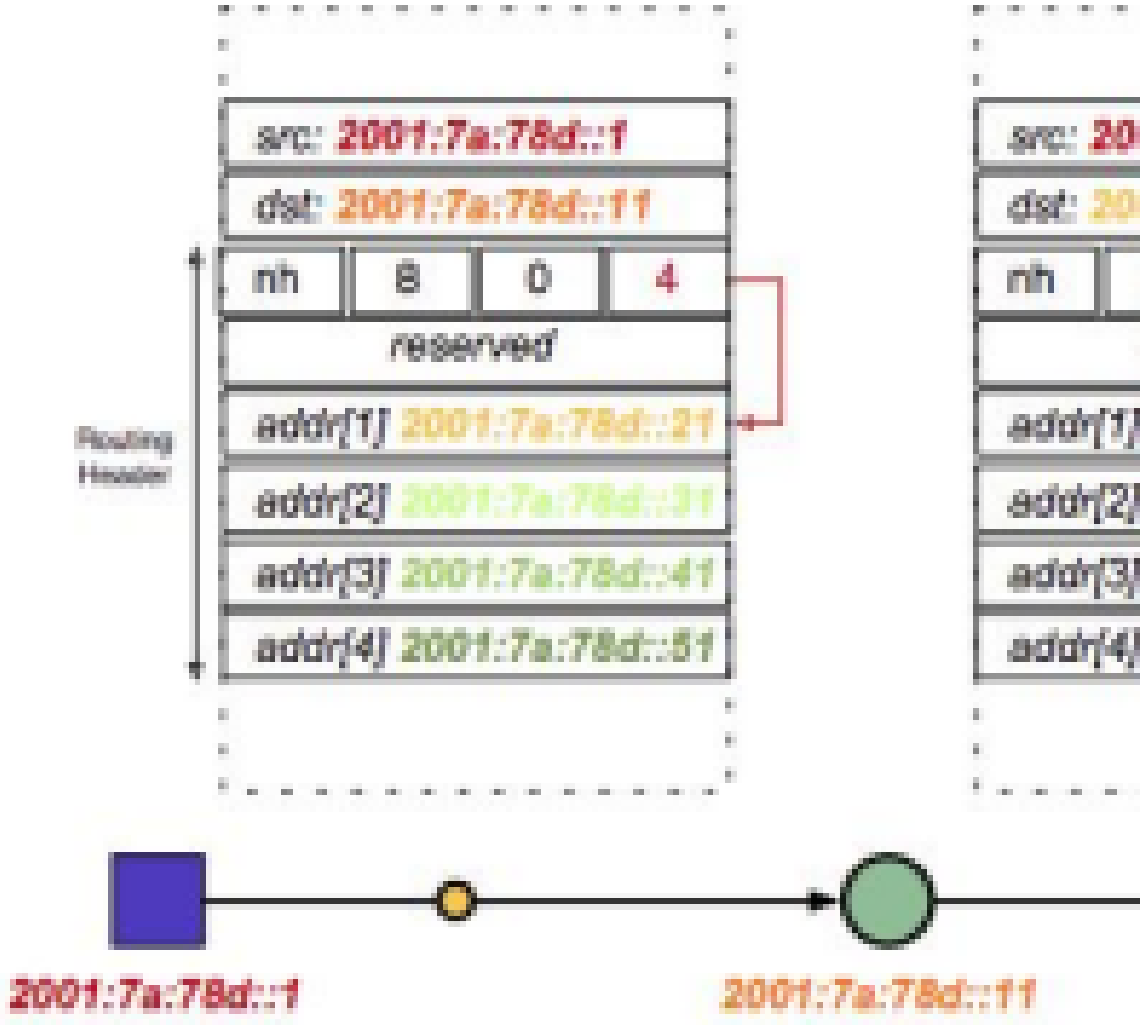


The control plane runs the dijkstra to find the shortest path from source to destination. But what if I don't want to use this path, but I want the traffic to pass specifically through some routers? This is called explicit routing: I want the traffic to pass specifically to R1 - R2 - R3 - R4 - R5. We use MPLS technology for this: I define a LSP and I install the different forwarding rules. with IPv6 we can do that without labels, using routing headers. The guy in charge of doing this is the source node, so it's called source routing.

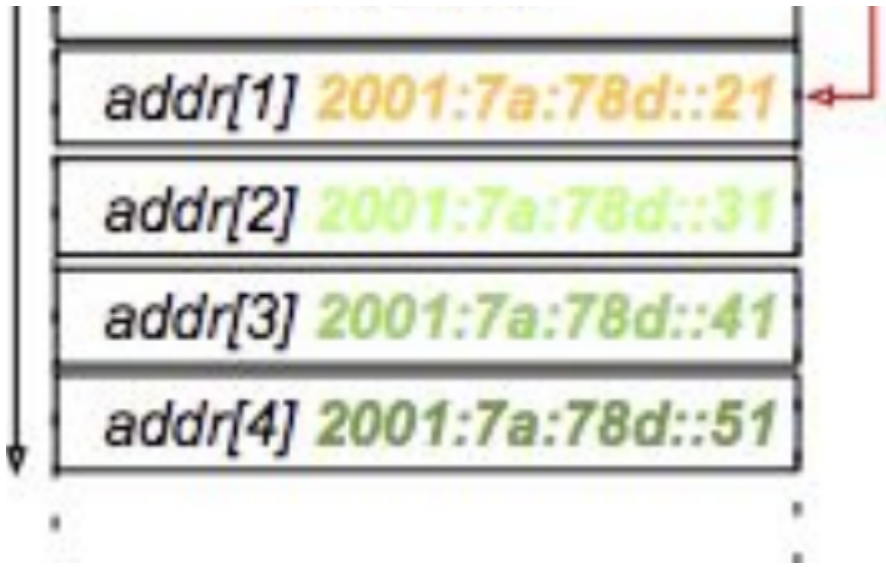
Assume Hs wants to use this source routing. We need to specify the destination + list of intermediate nodes we want to pass through. I write this list in the routing header: <R1, R2, R3, …, HD>.

Type 0 Routing Header:
Considering that each identifier is a 128 bit IPv6 address, this header is huge. How to compress this and not use all these bits? If I want to go from R1 to R3 I don't have to store R2 because I know that from R1 the shortest path to R3 is through R2

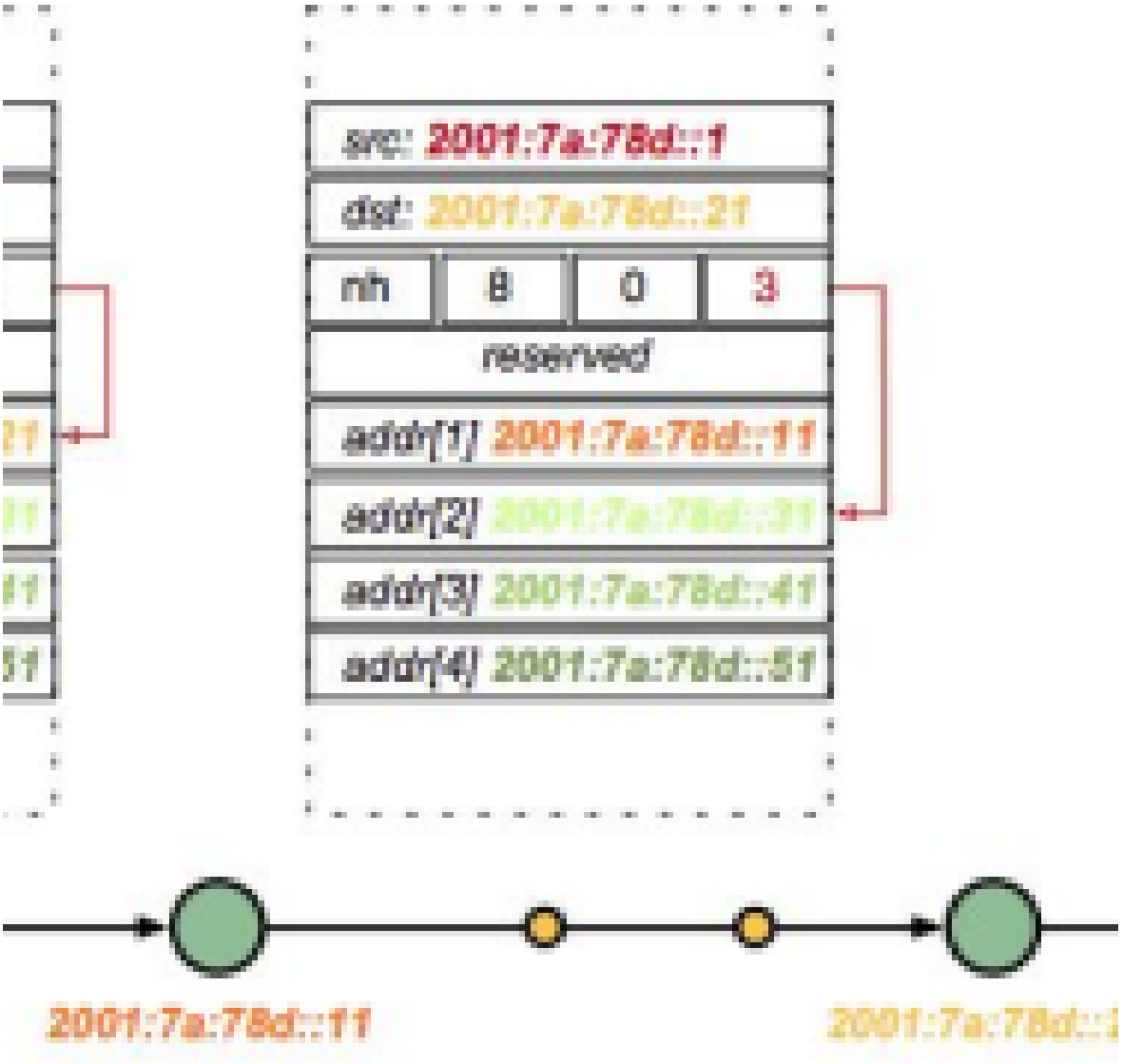
This is the path I want to follow, each circle represents an intermediate router. Small ones are modeling R1 and R2, so transit routers. R3 is a middle point, a node that we want to pass through, and the one we have to specify in the header list.



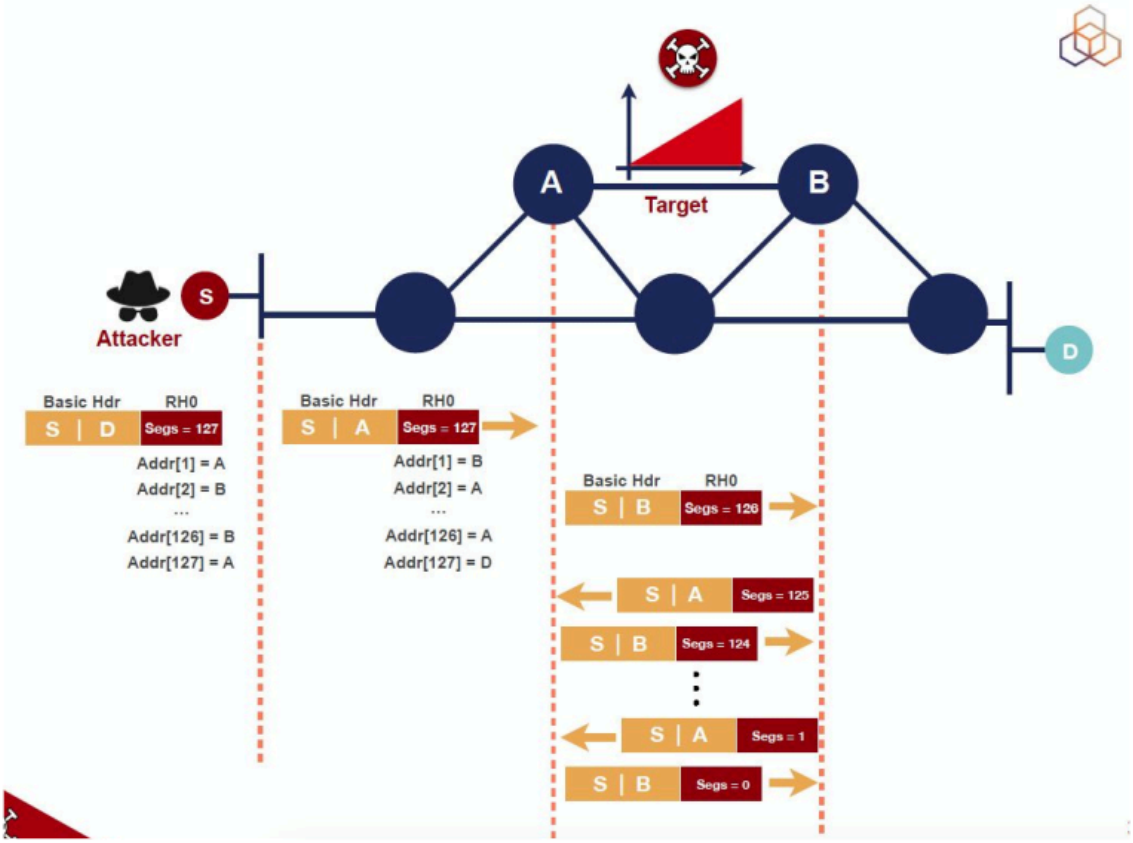
As shown above, the destination of the packet is the first middle point (R3), so not the actual destination. This way we can deliver the packet to the intermediate nodes we wanted. So I say to the Control Plane: Please find the shortest path (to HD) that passes through this intermediate node (R3 and the other middle point) in such a way that also passes through these little circles (R1, R2, …).



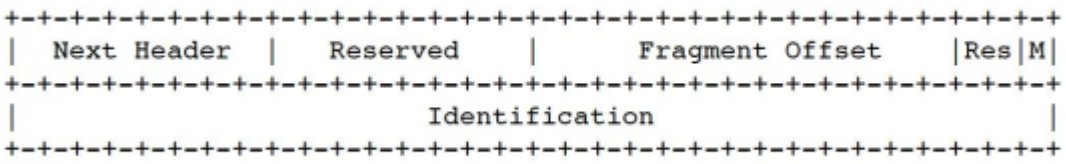
When the packet arrives in the middle point, the headers are inspected and updated, so the destination is changed with the first in the list:



Type 0 Routing Header is deprecated since it has problems with security: an attacker can make a DoS attack by just specifying A and B multiple times in the list of middle points to make the packet loop between A and B for a period. So an attacker can saturate the bandwidth with just few bits per seconds since it get an amplification

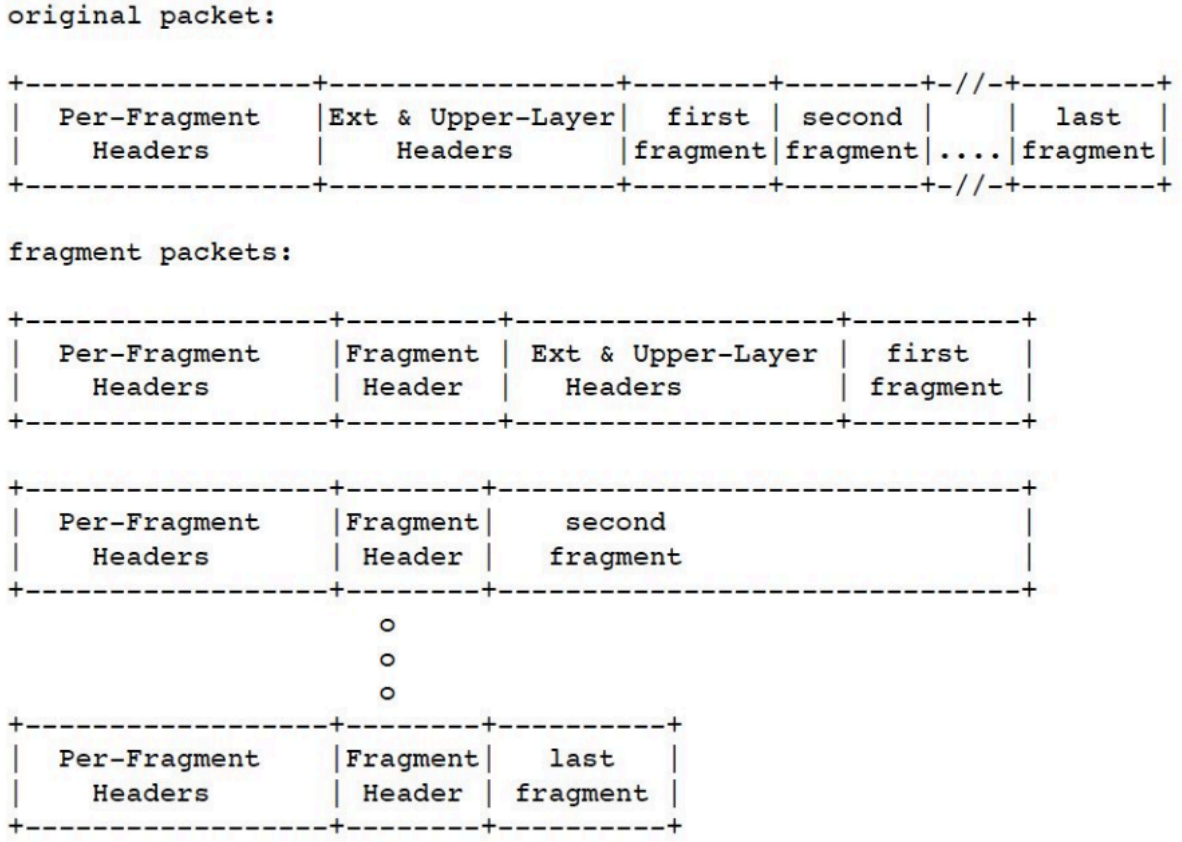


Fragment header:



Identification: specifies the fragments which belong to the same packet. Offset: specifies the position of the fragment w.r.t. to the original packet. M = 0 if it's the last fragment, 1 otherwise

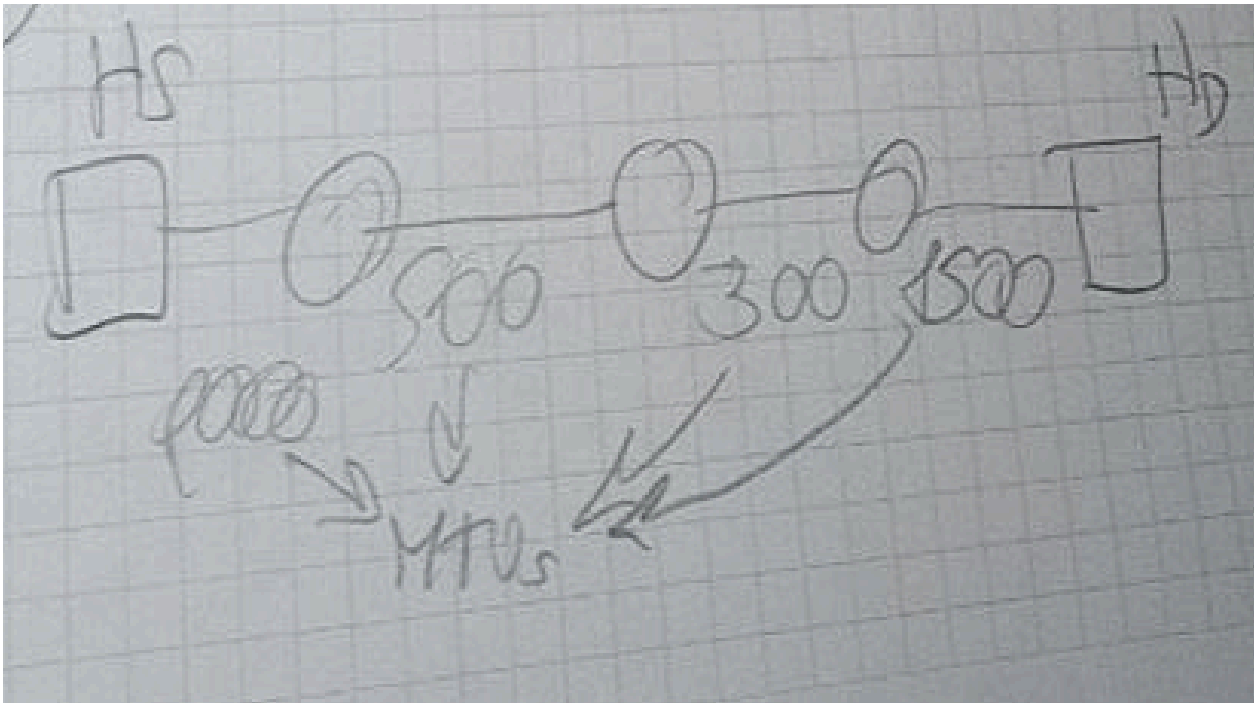
In IPv4 we copy the original header in every fragment. In IPv6 there are some headers that we don't need to copy. The ones that are per fragments: basic, hop-by-hop must be copied in each fragment. The headers that are inspected only in the destination, however, are present just in the first fragment.



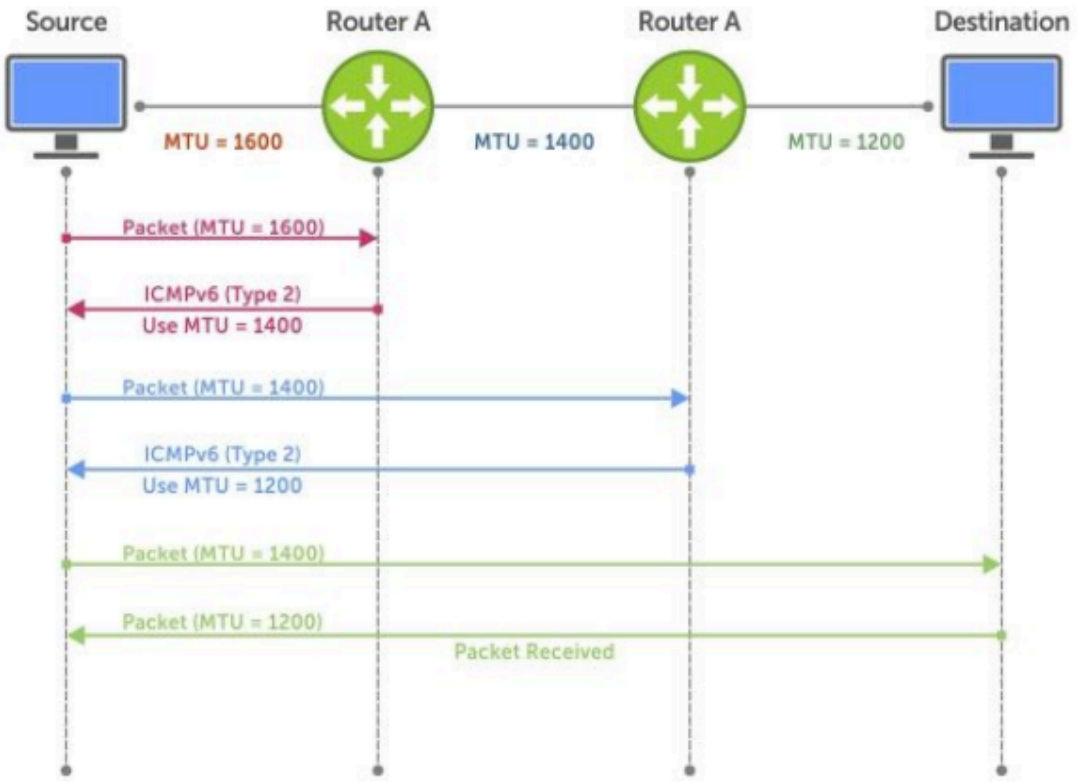
Fragmentation in Ipv4 fragmentation is performed in each single node, meanwhile IPv6 allows it only in the source nodes. So the source has to create the packets of the proper dimensions, specified by the Maximum Transmission Unit (MTU).

The pros is that there is less delay (since only the first and the last router has to operate) and less overhead (I don't have to carry the bits related to fragmentation if it is not needed), the cons are:

1. The need of Path MTU Discovery. Let's consider this case:



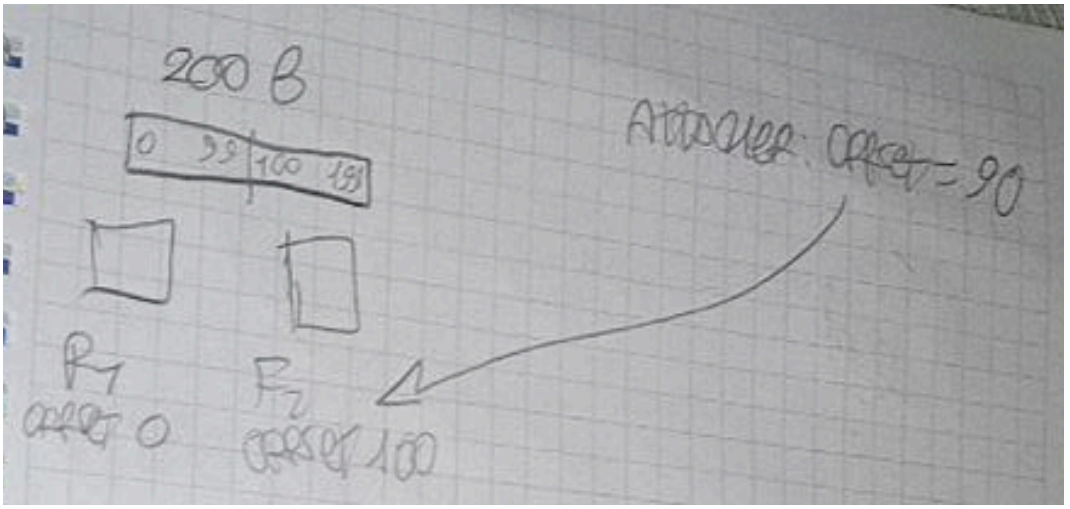
The source node is the one responsible for the creation of packets of the proper dimension. The other nodes will drop the packet if it's bigger than the MTUs. So we have to choose the smallest MTUs on the path, but how can the source node discover it? With the Path MTU Discovery



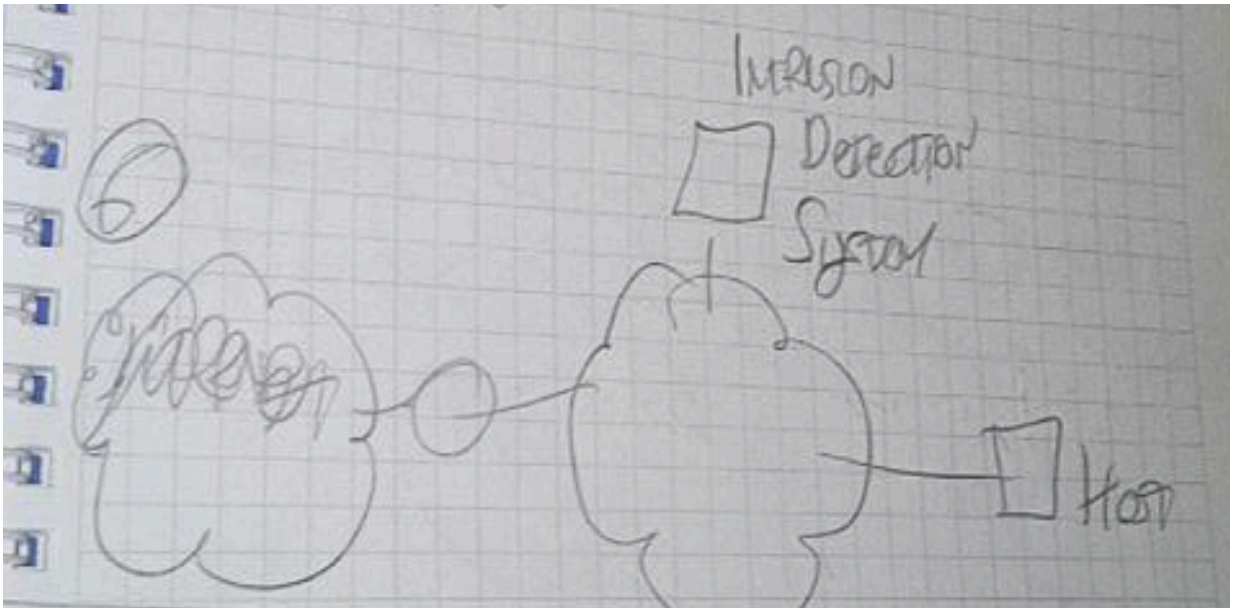
when you create the packet you assume MTU = 1600 is the smallest, but router A has MTU = 1400 so it discards the packet. ICMPv6 error code (packet too big) is delivered back to the source with the new MTU. In this way I create the smallest

amount of fragments I need to deliver the packet and I am sure it can reach the destination from the source.

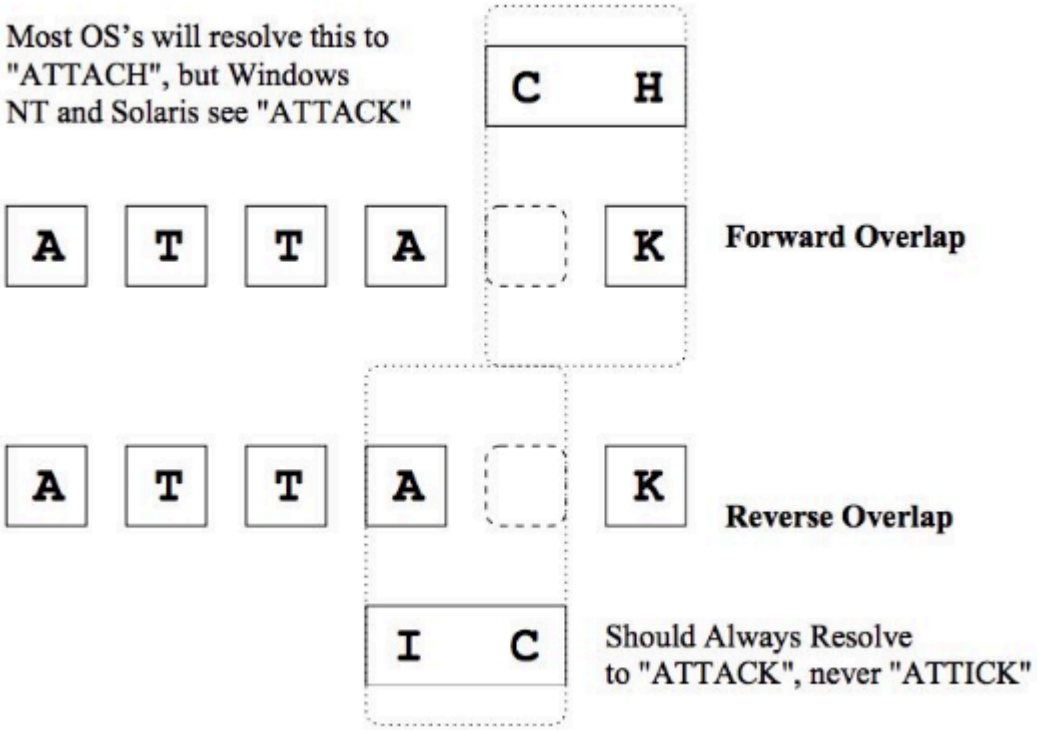
2. Security threats. Fragment Overlapping: there is no specification in the standard on what to do when two fragments overlap. Let's consider this case:



What should I do? Override the first in the second or vice versa? Nowadays, if there is an overlap we drop the packet. Fragment overlapping can be used by an attacker, so we need a way to detect it:



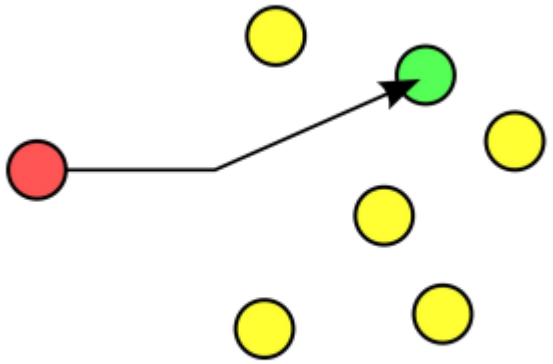
IDS receives the traffic and compares the different messages with a database of malicious messages, if we see that a specific message is malicious we raise an alarm. An attacker that understands how the target OS and the IDS work in this context can obscure his attack



Ipv6 Addresses:

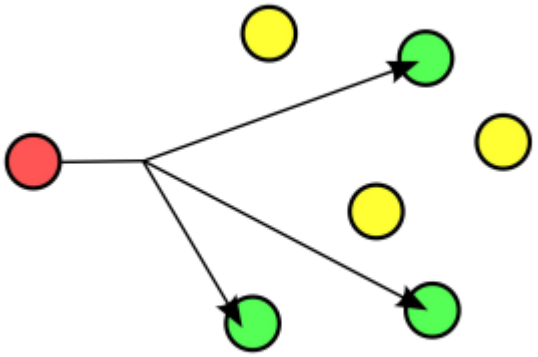
There are three types of addresses:

1. Unicast: one-to-one communication paradigm



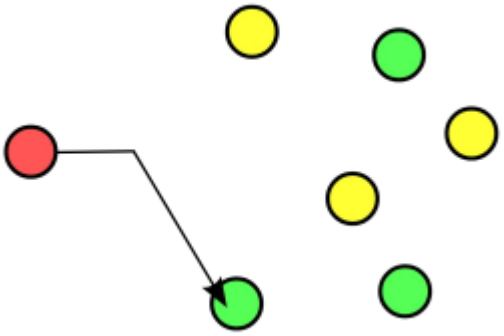
The red node is the source, the green node is the destination. most of the communications we have are unicast.

2. Multicast:



There is a single source and multiple destinations. This is useful in streaming applications: transmitting a football match, it's real time, I create a stream that is of interest for some of you. Broadcast is a specific case of Multicast, in which every yellow node is green.

3. Anycast:



There are potential multiple receivers, but at least one receives the message. E.g. you want to get a service and it's provided by multiple nodes, e.g. DNS, you just need one DNS to respond to you (the closest one). Anycast addresses are taken from the unicast address spaces.

Text Representation of IPv6 Addresses

In IPv6 we don't use dotted decimal notation: we use hexadecimal representation: 8 groups of 16 bits. A 128 bit packet is divided into 8 blocks each one with 16 bits, and since each block is in hexadecimal, we use 4 bits for each block. How to make it shorter: we don't have to specify the leading zeros.

2001:DB8:0:0:8:800:200C:417A

e.g. here, DB8 is actually 0DB8, also 0 is actually 0000, and so on. We can also use the notation :: to compress a long list of zeros.

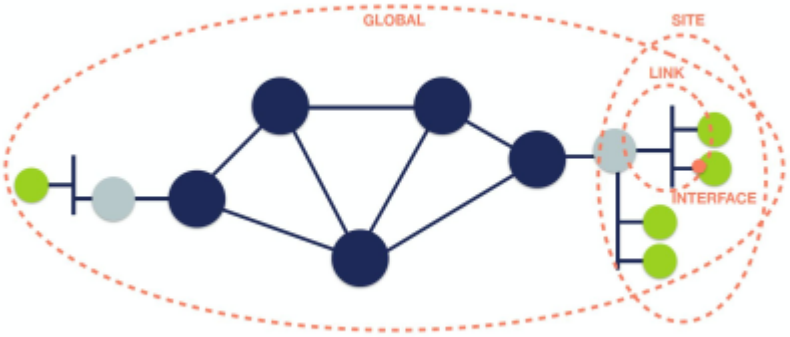
2001:DB8:0:0:8:800:200C:417	2001:DB8::8:800:200C:417	*a unicast address*
FF01:0:0:0:0:0:101	FF01::101	*a multicast address*
0:0:0:0:0:0:1	::1	*the loopback address*
0:0:0:0:0:0:0	::	*the unspecified address*

But we can use the :: notation at most 1 time: e.g. 2:0:0:0:5:0:0:8

i can represent it this way: 2::5:0:0:8
i can represent it this way: 2:0:0:05::8
but not like that: 2::5::8 because we don't know how many blocks we have in the first part and in the second, it can me 2 and 2, or 4 and 1

IPv6 Address Prefix:
The smallest network that you can specify in IPv6 is /64. In 2001:0DB8:0000:CD30:0000:0000:0000:0000/60 the address is 2001:0DB8:0000:CD and the other part is the prefix.

There exist different types of addresses based on the scope:



Address type	Binary prefix	IPv6 notation
Unspecified	00...0 (128 bits)	::/128
Loopback	00...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-Local unicast	1111111010	FE80::/10
Global Unicast	(everything else)	

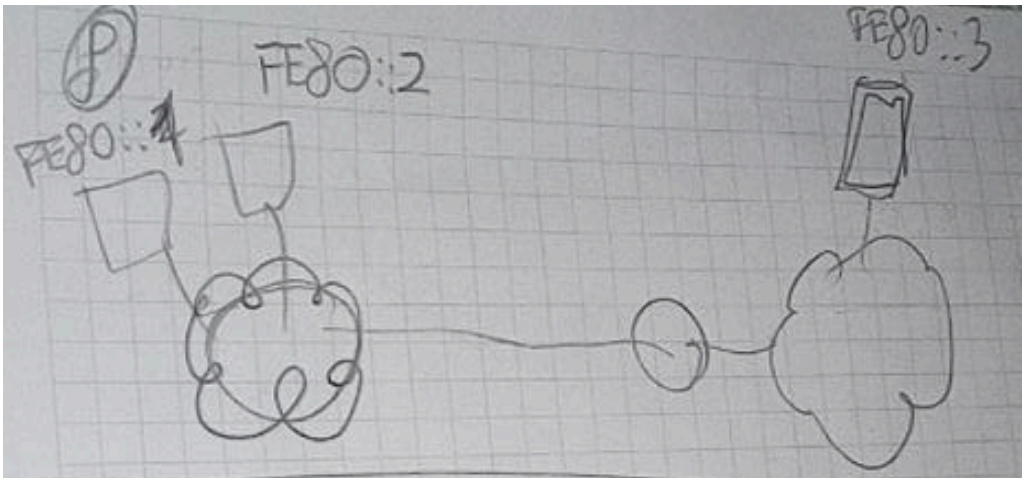
1. Link Local addresses are used to communicate with hosts connected to the same Local Network (on the same link). This address is assigned to the card by the OS automatically
   they are used for:
   a. automatic address configuration
   b. neighbor discovery
   c. when no routers are present

   Routers must not forward any packet with link-local source or destination addresses to other links. This is the format

10 bits	54 bits	64 bits
1111111010	0	interface ID

   with fe80 (1111111010) that identifies that the scope is Link Local

2. Global addresses:
   Let's consider this situation:

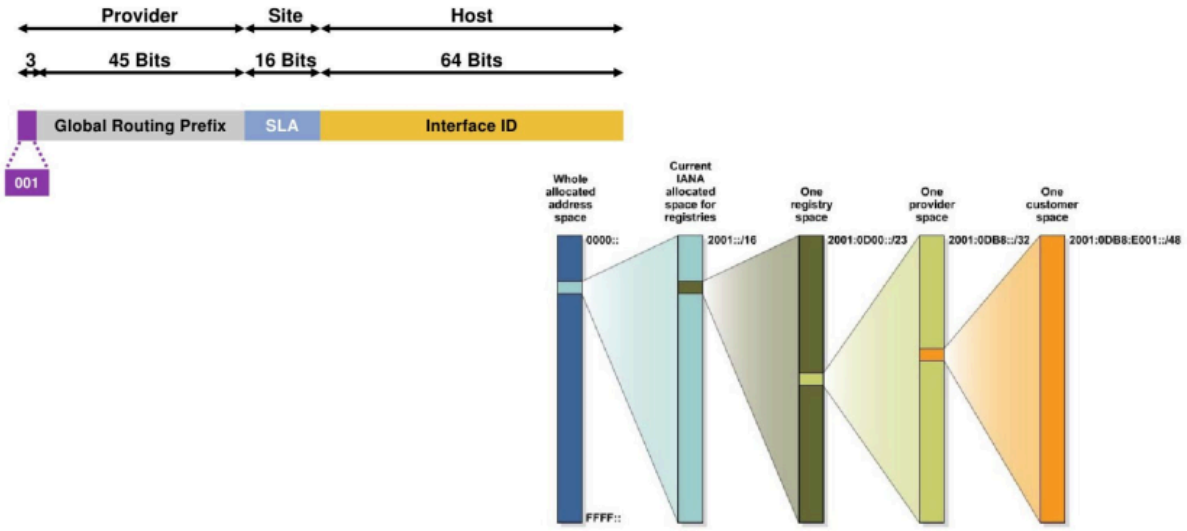


   To communicate with the other net, I can't use FE80::3, I have to use a global address. This is the global address structure:

n bits	m bits	128-n-m bits
global routing prefix	subnet ID	interface ID

   where:
   - The n bits global routing prefix is the value assigned to a site
   - The m bits subnet ID identify the link in a site
   - The 128-n-m bits

   For the moment this is the situation of usage of IPv6 addresses: we are just using Current IANA allocate space for registries (in cyan).



3. SITE is deprecated

Let's assume we are a mid-sized company with offices and data centers across the USA. ISP assigned it the block 2001:db8:1234::/48[5]. How to allocate this block across the enterprise?

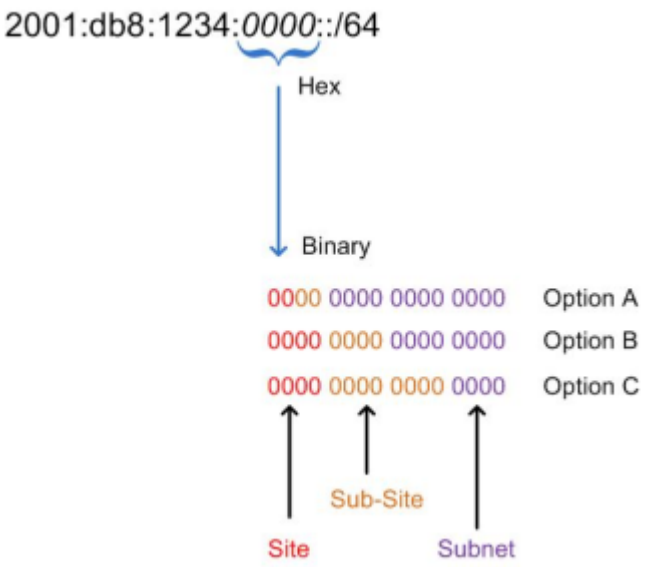
n bits	m bits	128-n-m bits
global routing prefix	subnet ID	interface ID

Assigning a /64 to the prefix (n + m), since we received a /48 (n) prefix from the ISP, leave us with 64 - 48 = 16 bit for subnetting, as well as the last 64 for the interface ID. 16 bit for subnetting means half of the entire IPv4 space just for subnets in our organization.

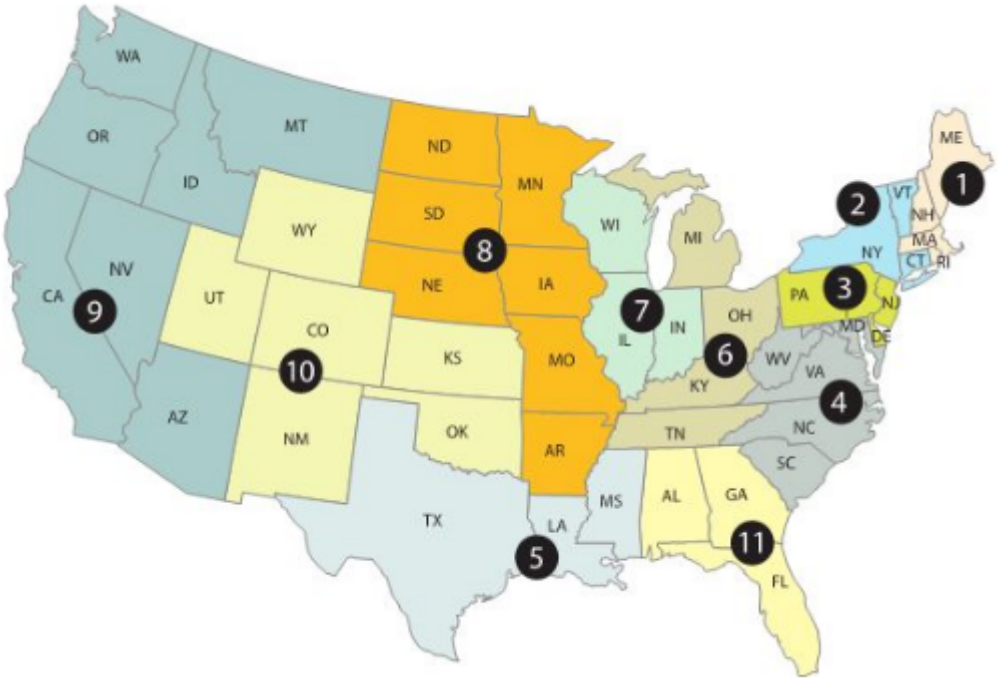
2001:db8:1234:*0000*::/64

We should define at least one Site ID and possibly a sub-Site ID. We can do that in 3 way:

[5] The leading practice is to receive at least a /48 prefix from the ISP, so to have 128 - 48 = 80 -> 280 bit to manipulate, more than the entire IPv4 space only for our organization!



We could use option B, that gives us $2^4 = 16$ sites (geographic region of the country), $2^4 = 16$ sub-sites (city in the region) and $2^8 = 256$ subnets per site.



- Site 0 - 2001:db8:abcd:0000::/52 (for future use)
- Site 1 - 2001:db8:abcd:1000::/52
- Site 2 - 2001:db8:abcd:2000::/52
- Site 3 - 2001:db8:abcd:3000::/52
- ...
- Site 8 - 2001:db8:abcd:8000::/52
- Site 9 - 2001:db8:abcd:9000::/52
- Site 10 - 2001:db8:abcd:a000::/52 (for future use)
- Site 11 - 2001:db8:abcd:b000::/52 (for future use)
- Site 12 - 2001:db8:abcd:c000::/52 (for future use)

- site 1
  - Boston - 2001:db8:abcd:1100::/56
  - others for future use
- site 2
  - New York City - 2001:db8:abcd:2000::/56
- site 3
  - Newark - 2001:db8:abcd:3f00::/56
- site 8
  - Omaha - 2001:db8:abcd:8000::/56
- site 9
  - San Francisco - 2001:db8:abcd:9100::/56
  - Seattle - 2001:db8:abcd:9200::/56

For instance, let's start with our offices in Newark. The block 2001:db8:1234:0000::/64 is assigned here as 2001:db8:1234:3f00::/64 since:

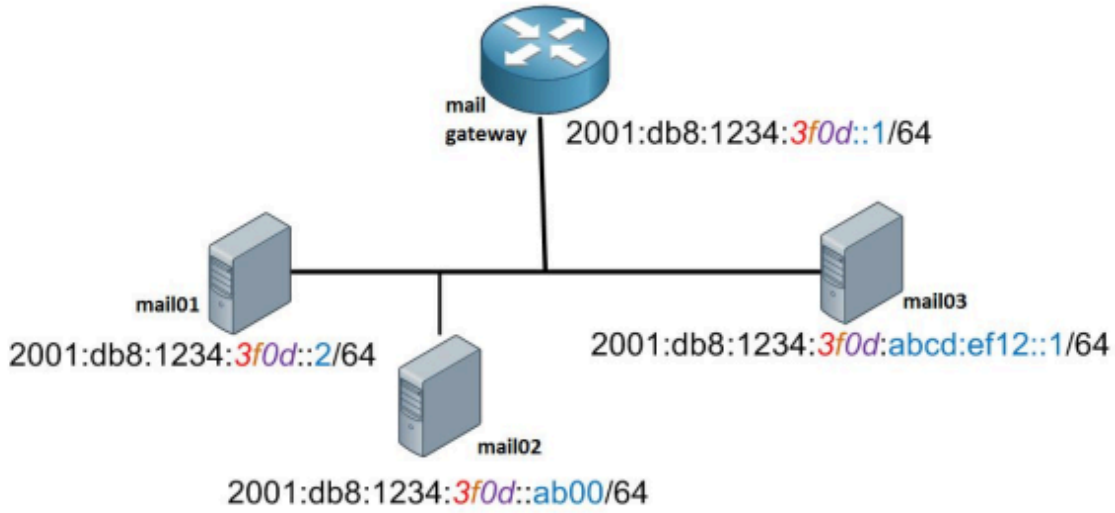
3 -> 0011 (for the site 3, the geographic region)
f -> 1111 (for the sub-site, the city Newark)
00 -> 0000 0000 (for the subnets)

With the last two hex digits, that are 8 binary digits, we can address 2^8 = 256 specific services of our Newark offices, that are our subnets:

- Firewall Outside: 2001:db8:abcd:3f00::/64
- Webservers: 2001:db8:abcd:3f01::/64
- Database Servers: 2001:db8:abcd:3f02::/64
- ....
- Mail Servers: 2001:db8:abcd:3f0d::/64
- ....
- Management: 2001:db8:abcd:3fee::/64
- Loopbacks: 2001:db8:abcd:3fff::/64

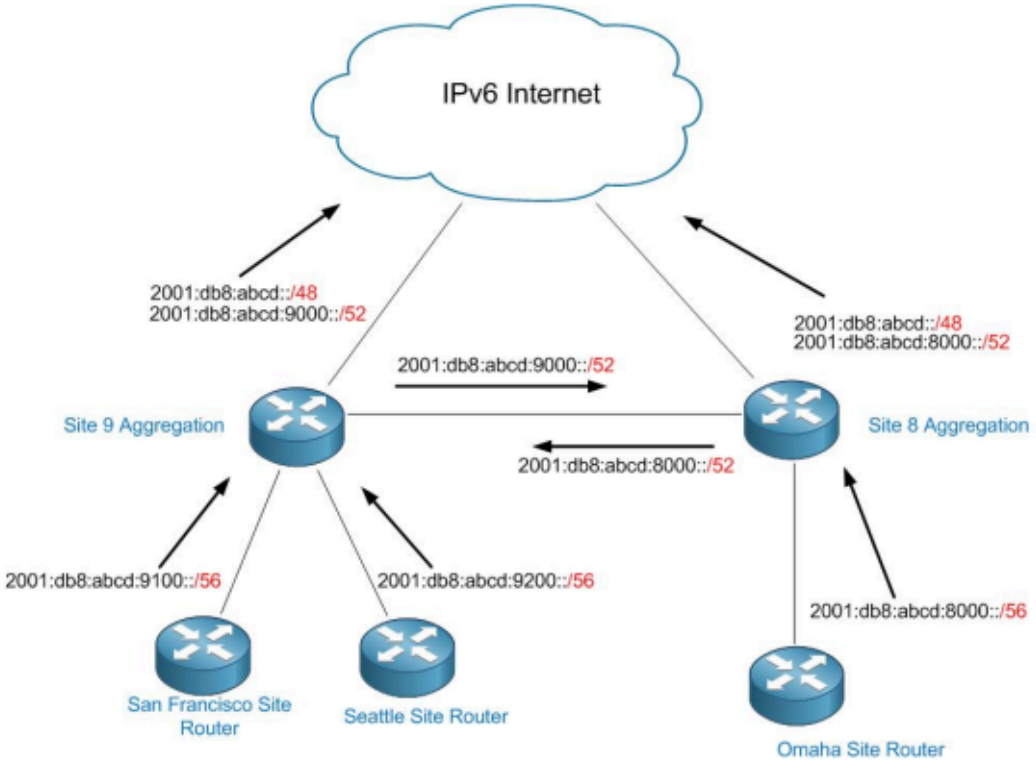
Now we can assign addresses to the specific hosts, using the remaining blocks which we compressed with the :: notation. We can assign 2^{64} hosts for each subnet, namely for each

service. For instance, for Mail Servers:



Mail Servers: 2001:db8:1234:3f0d::/64

And so the routing from the Sub-Site Routers to the IPv6 Internet is like that:

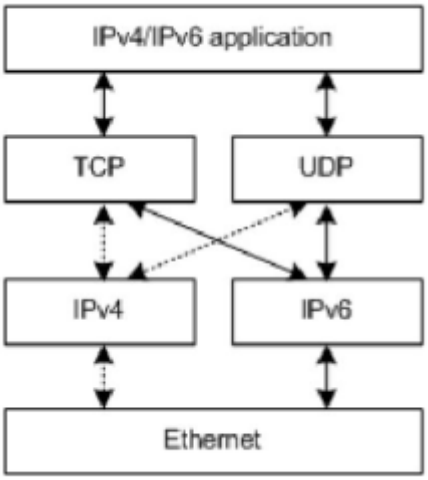


With /48 which are addresses that the ISP provided, /52 (first hex bit fixed) that are Site addresses (geographic region of USA) and /56 (first and second hex bit fixed) that are Sub-Site (City in each Geographic Region)

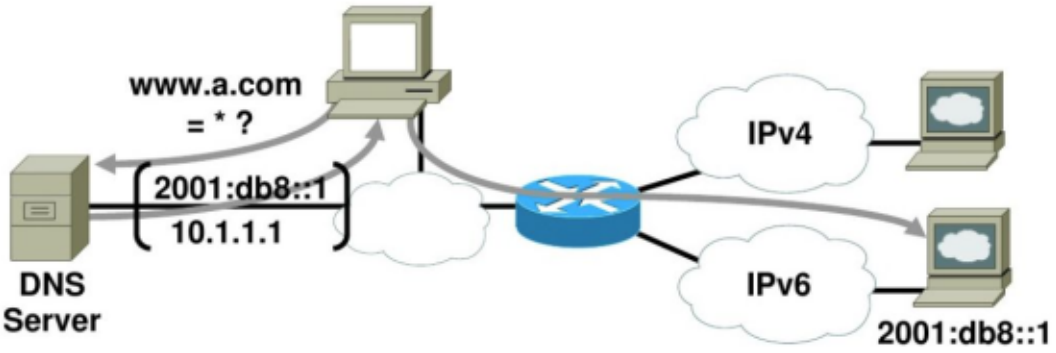
IPv6 Transition Mechanism

The key to the transition from IPv4 to IPv6 is compatibility with the installed base of IPv4 hosts and routers. There are two mechanism:

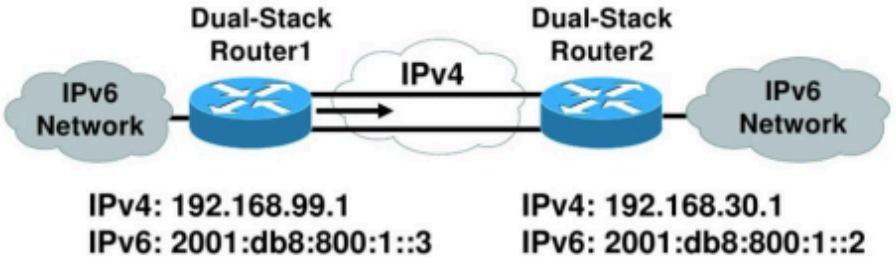
1. Dual IP Layer (Dual Stack): complete support for both IPv4 and IPv6 in the servers, in the DNS and in the routers! Not in the clients!.



This means that nodes may be configured with both IPv4 and IPv6 addresses (e.g. DHCPv4 + SLAAC) and that we use a new resource record type “AAAA” for IPv6 addresses, so nodes must be able to deal with IPv4 “A” records as well as IPv6 “AAAA” records.



2. Tunneling of IPv6 over IPv4: establishing p2p tunnels by encapsulating IPv6 packets in IPv4. This way, while the IPv6 infrastructure is being deployed with its time, the existing IPv4 infrastructure can also carry IPv6 traffic. This tunneling can be used:
    a. Router-to-Router
    b. Host-to-Router
    c. Host-to-Host
    d. Router-to-Host



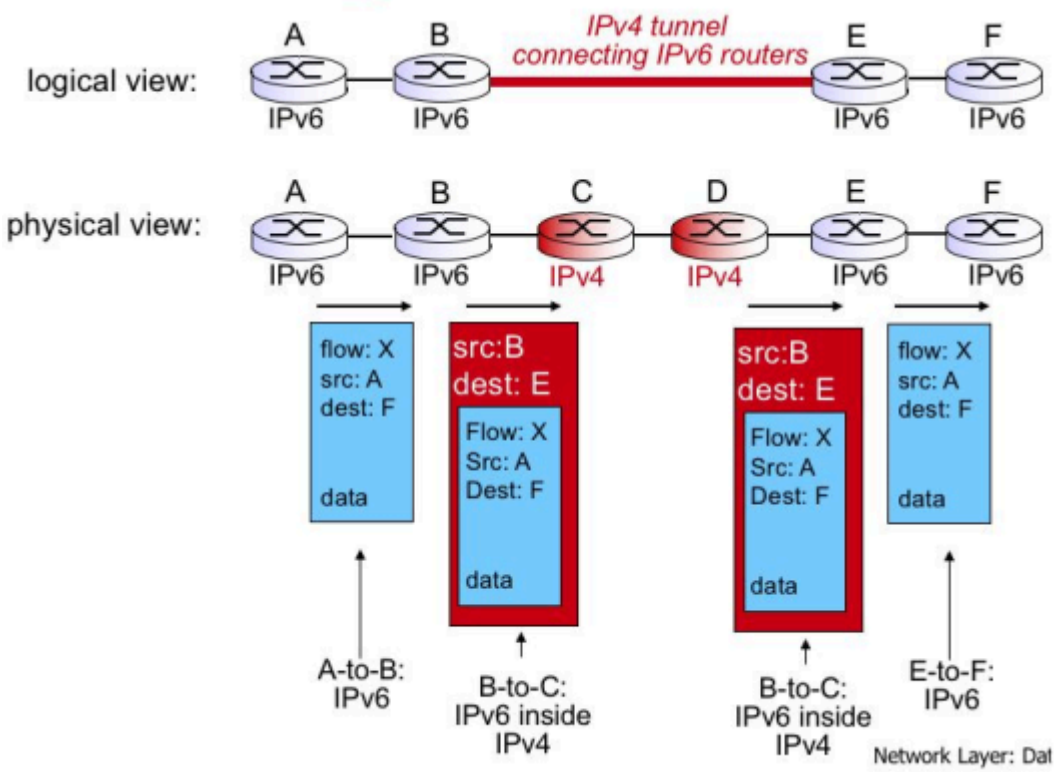

```
router1#

interface Tunnel0
 ipv6 enable
 ipv6 address 2001:db8:c18:1::3/127
 tunnel source 192.168.99.1
 tunnel destination 192.168.30.1
 tunnel mode ipv6ip
```

```
router2#

interface Tunnel0
 ipv6 enable
 ipv6 address 2001:db8:c18:1::2/127
 tunnel source 192.168.30.1
 tunnel destination 192.168.99.1
 tunnel mode ipv6ip
```

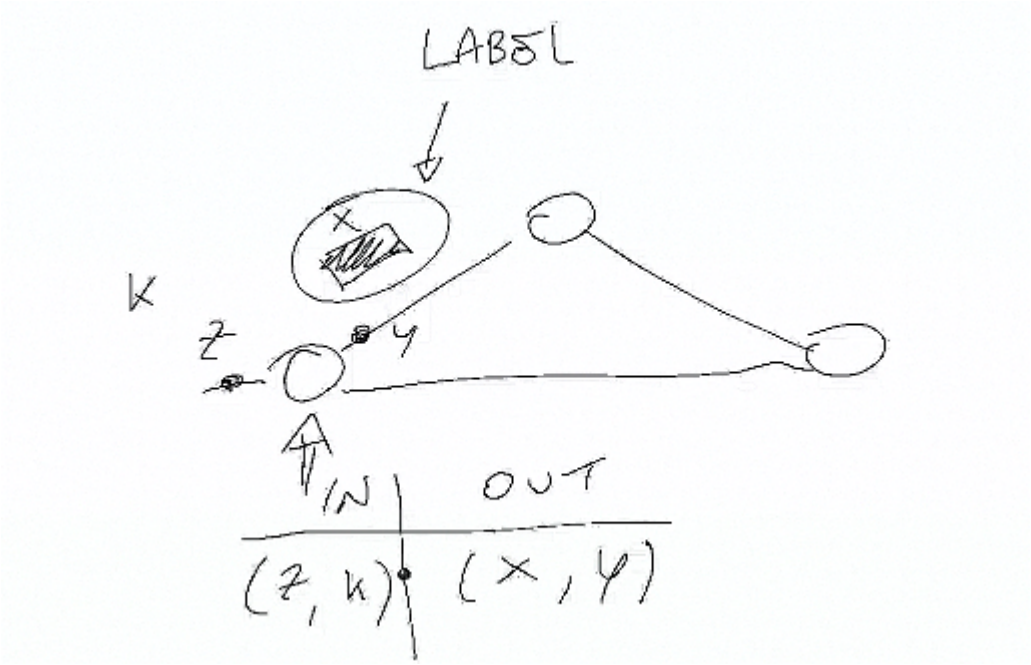
The entry node of the tunnel creates an encapsulating IPv4 header and transmits the encapsulated packet. The exit node receives the packet, reassembles it if needed, removes the IPv4 header and processes the received IPv6 packet.



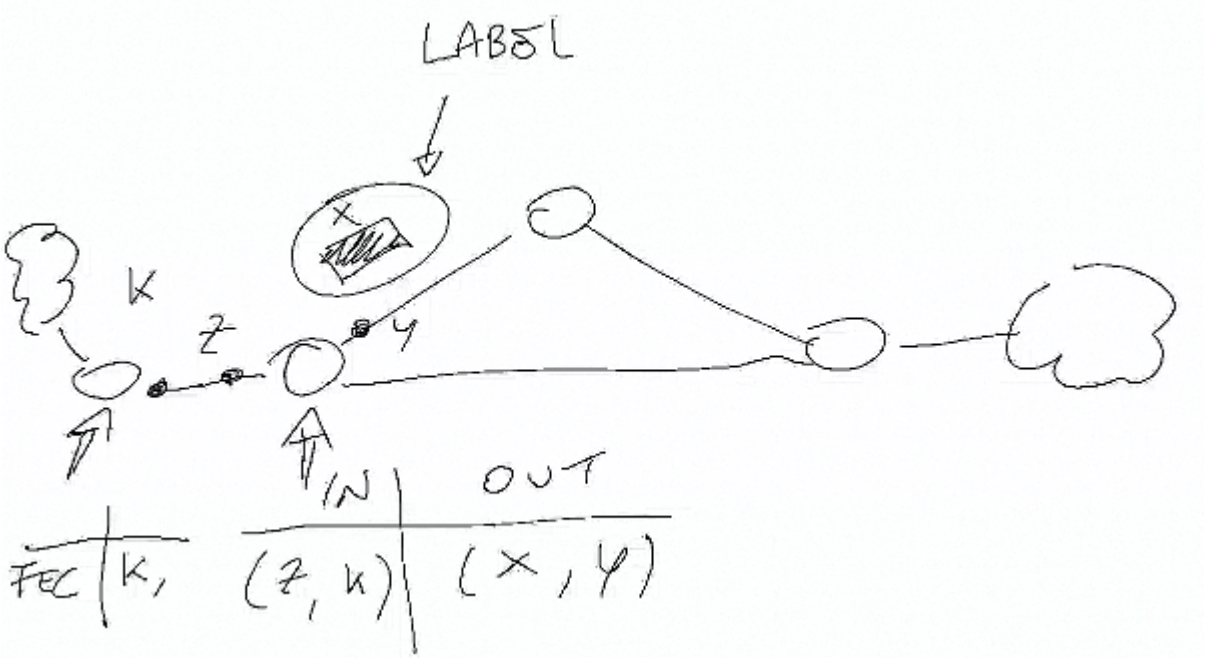
Segment Routing

27/10 and 30/10/23

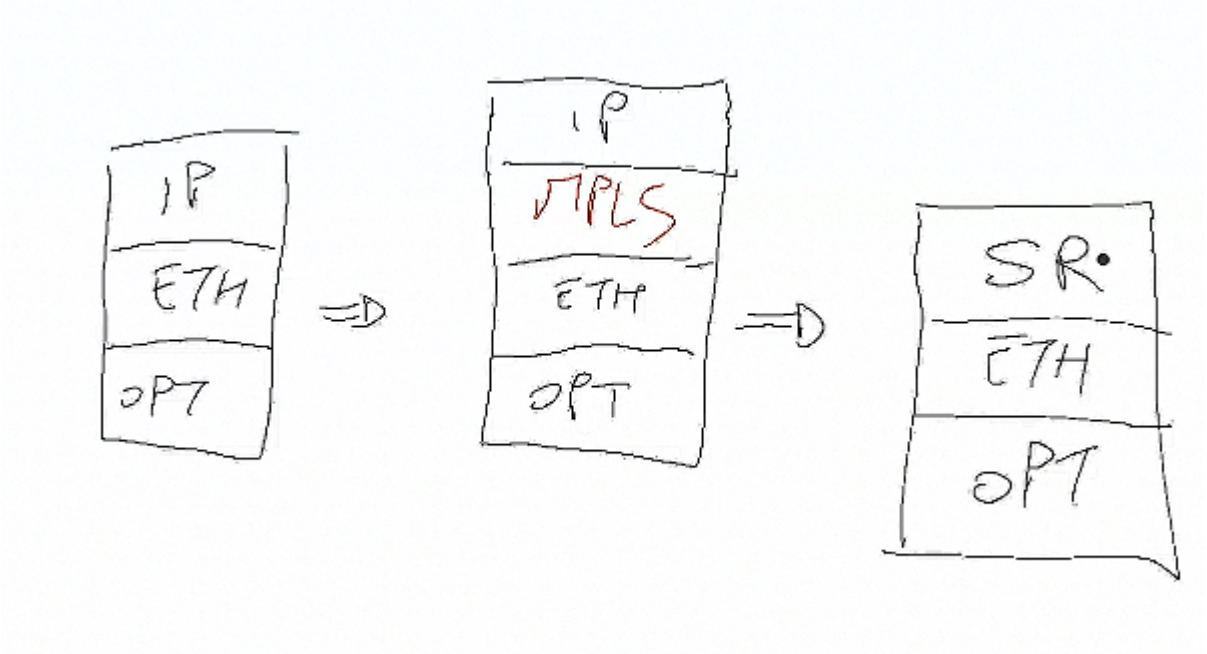
Why Segment Routing:
MPLS is expensive in the number of LSP created, each single LSP requires a setup with LDP and also LSP needs to be constantly refreshed, so there is a lot of overhead. Also the troubleshooting part is complex in MPLS: when you inspect a packet you just see a label, that has local validity, since the label could be used in multiple links, so you don't know the source, the destination address or which LSP is carrying the packet. What you have to do to reconstruct the path is to check the node that emitted the packet and inspect the Label Switching Table:



Now I know the previous node and I can inspect it, doing the same thing recursively, eventually reconstructing the history, until finding the FEC the packet belongs to.

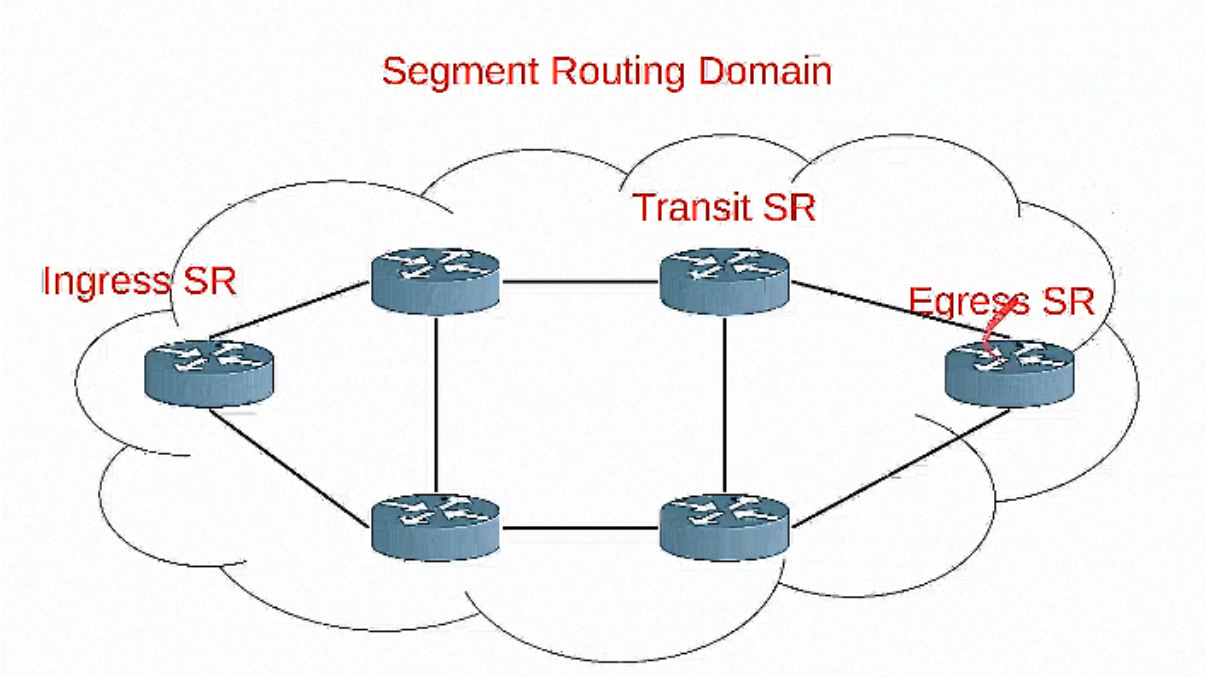


We have to do that for each packet, so it's very long. For this and many other reasons we remove the MPLS stack and we add segment routing.

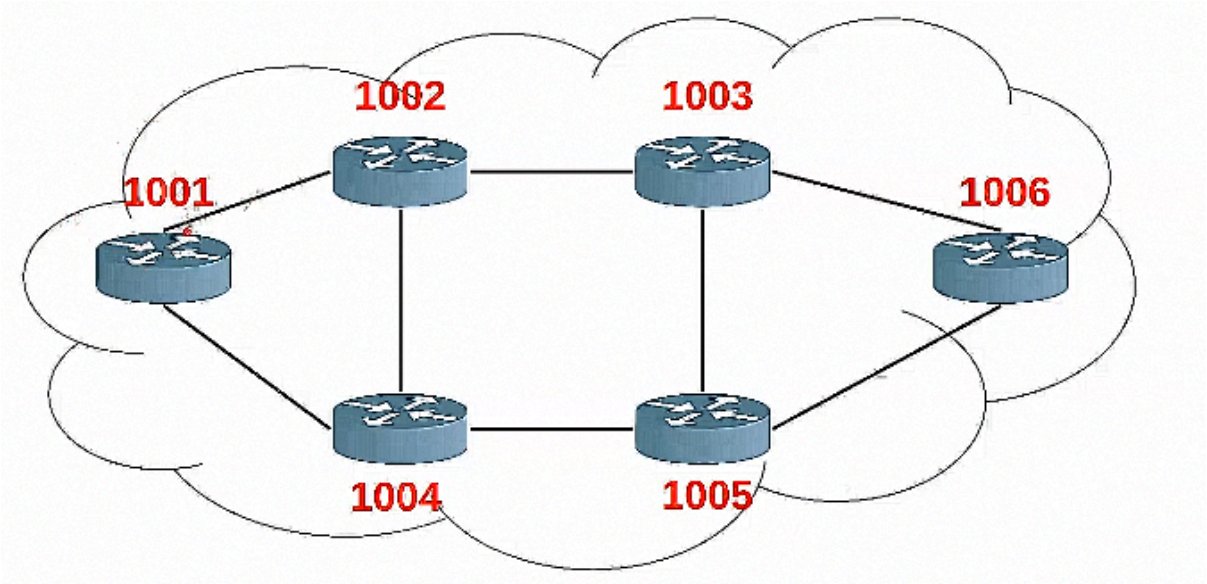


SR solves the problems of MPLS and keeps the benefits of MPLS: VPN, Traffic Engineering and resilience. SR supports the source routing paradigm.

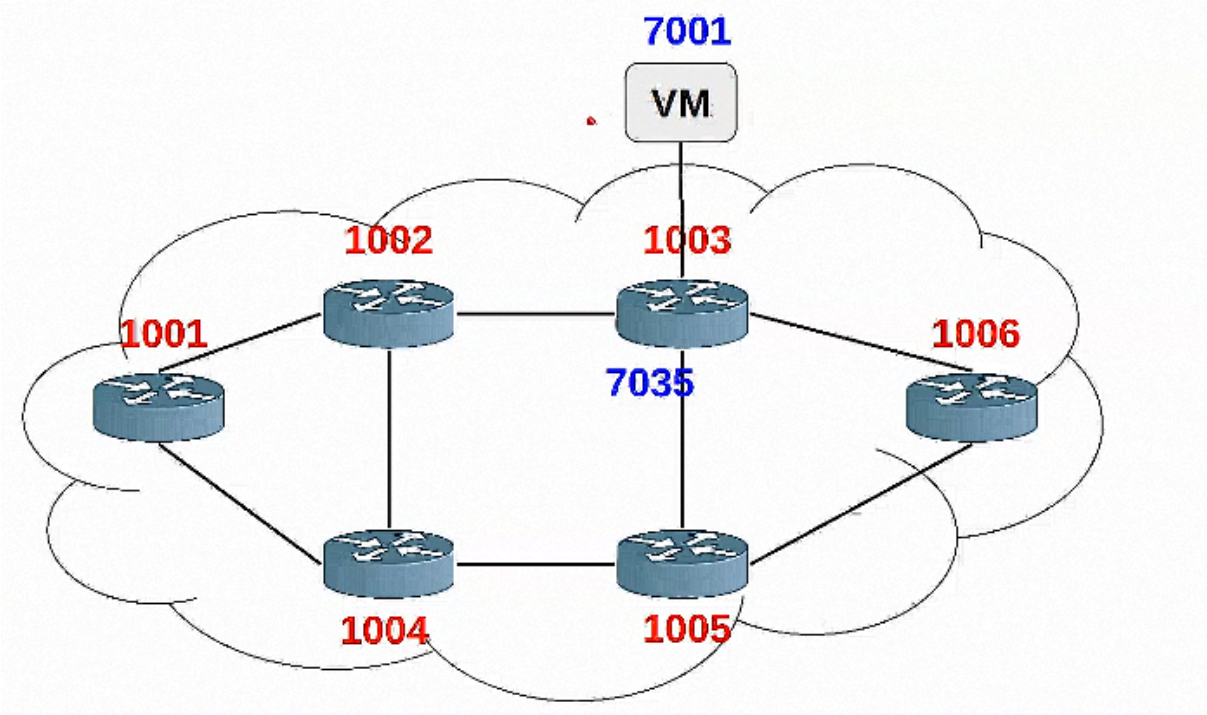
Segment ID (SID) and Segment List:
Let's consider this example:



we can think that attached to the ingress and egress there are customer nodes. Let's assign an identifier to every single node, called SID



I can assign identifiers also to other entities, like interface 7035 or service accessible through a specific node, e.g. VM 7001



The SIDs represent instructions. An instruction can be:

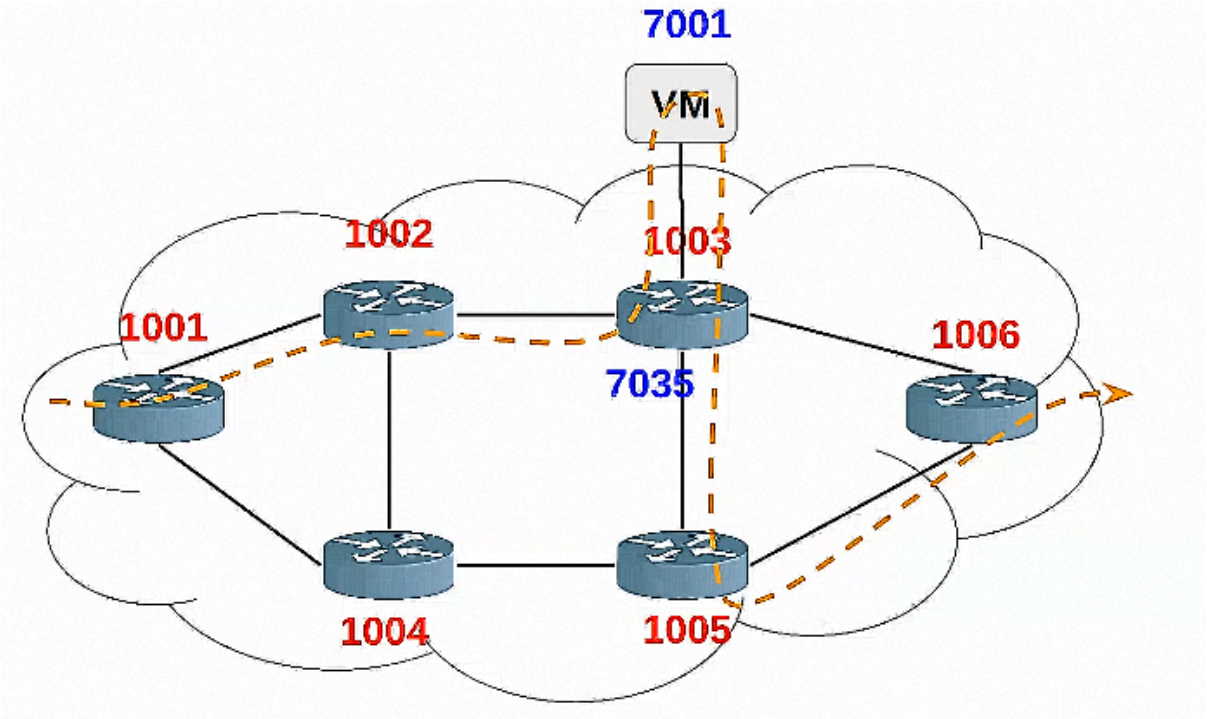
1. Topological. For example, the identifier of the node or of the interface is a topological instruction.
2. Service-based. The VM is a service-based instruction. The 7001 function could be everything, like switching a light bulb or reading the temperature, not only a VM. So it's not a simple topological instruction, but can be everything.

Moreover, the instruction can be:

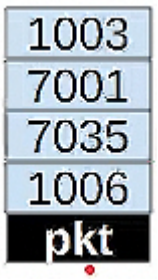
1. Local. An instruction is local when only a specific entity can perform the instruction. If I put in the packet the instruction "forward this packet out of the interface 7035" only the device in which the interface is installed can do it, so 1003. Why 7035 is local: because the interface is accessible only from 1003.

2. Global. In our case the red labels are global instructions, e.g. if 1002 receives the instruction 1005, it can send the packet to 1005, as every other node in the network.

Let's assume we want to follow this path:

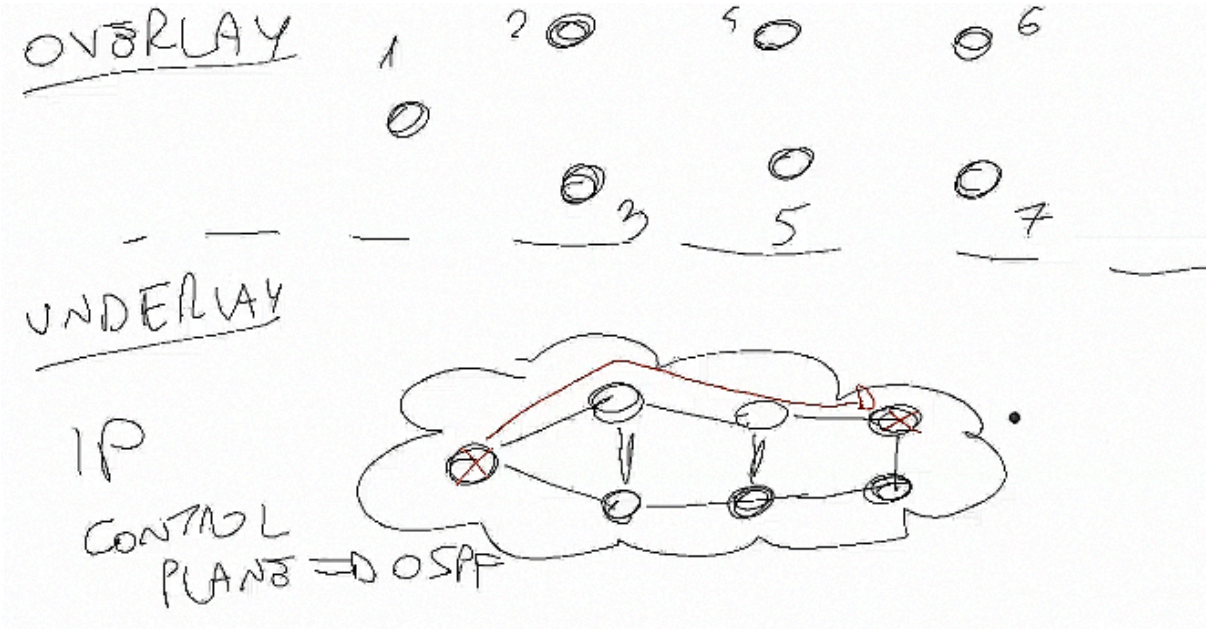


This path is unfeasible with IPv4 or IPv6, since we cannot decide the path (explicit routing) and also there is a loop in 1003, so 1003 performs two different actions with the same packet. With SR we can do that. We need this *Segment List*:

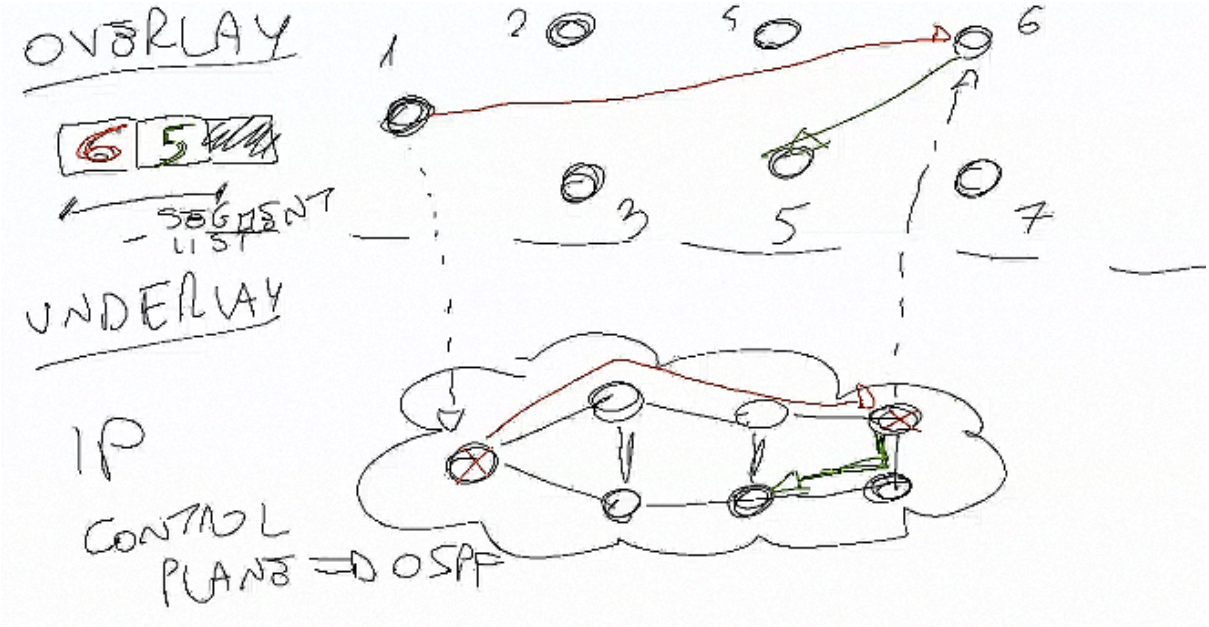


But this doesn't tell us, from 1001, what path should I use to go to 1003. We will use the shortest path since it's the one that we know. We know it because SR it's an overlay architecture that is built on top of an underlay network that is IP, so we have an IP Control Plane that runs OSPF. So I am sure that each node knows how to go from one node to another.

Let's make an example of this:



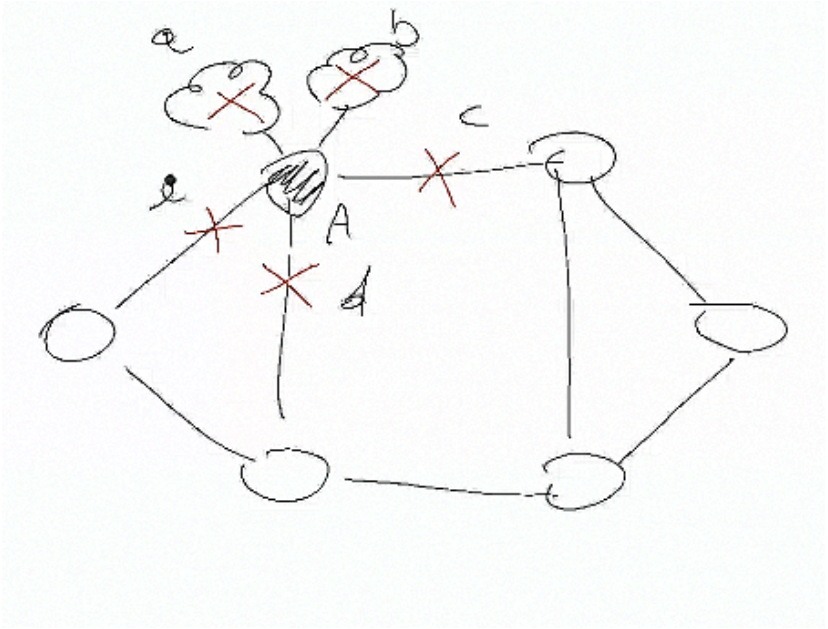
Imagine we want to visit and leave at 5. The node 1 asks the underlay to deliver the packet. Once we arrive at 6 the packet is sent to the overlay once again and 5 will be reached, and so on and so forth. From an overlay perspective we just did 1 hop but from an underlay perspective more than one.



As you can see, there are a lot of similarities between SR and the routing header. In fact, this is the theoretical explanation, practically this SR is implemented in IP.

SR Control Plane:
In the SR architecture we can have a:

1. Distributed Control Plane. In which segments are allocated and signaled by OSPF (or BGP) in order to make each node aware of the other node SID and each node individually computes the source-routed policy



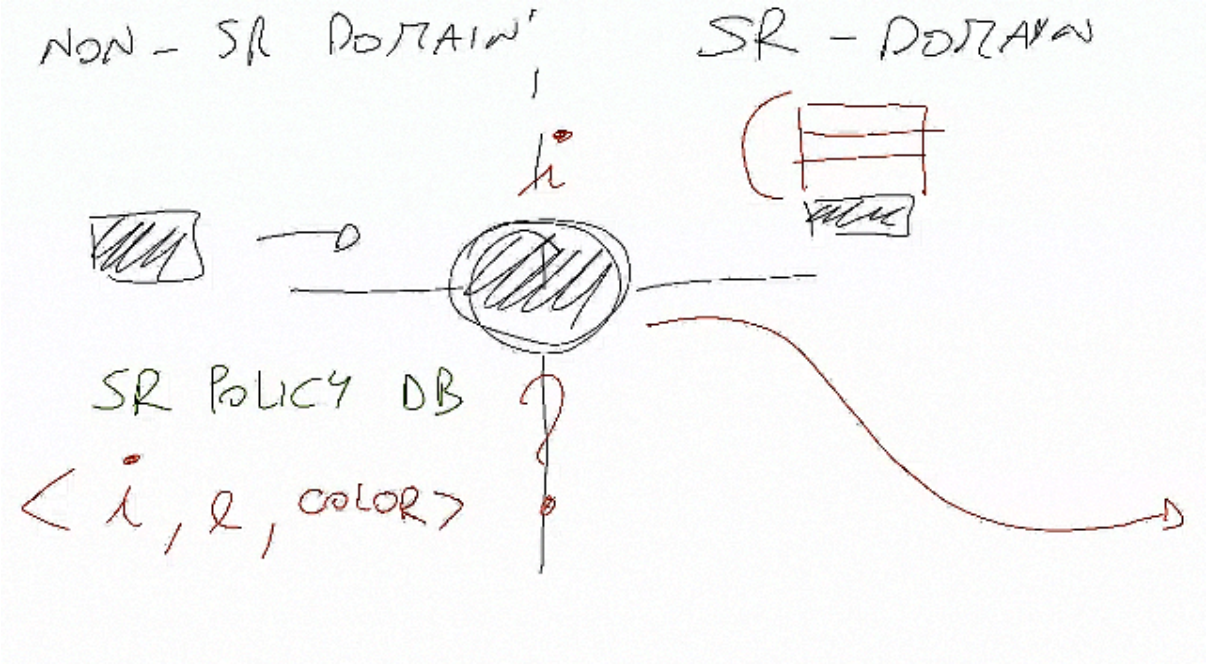
Let's make an example of the OSPF protocol. In this example router A is connected to 5 subnetworks. It creates a LSA (Link States Advertisement) that contains the identity of the router A and the list of networks that are connected to A.

The router sends this message to all its neighbors. Each router has a topology database, once a router receives the LSA it inserts it into the DB. So it starts building the vision of the network. Also routers that are not connected directly to A must receive its LSA, so we do *flooding*, with which routers directly connected to A deliver its LSA to other routers. Let's say that A has also some SIDs, for the interfaces, for the services etc. I want to inform the other routers also about this. I just have to put them in the LSA.

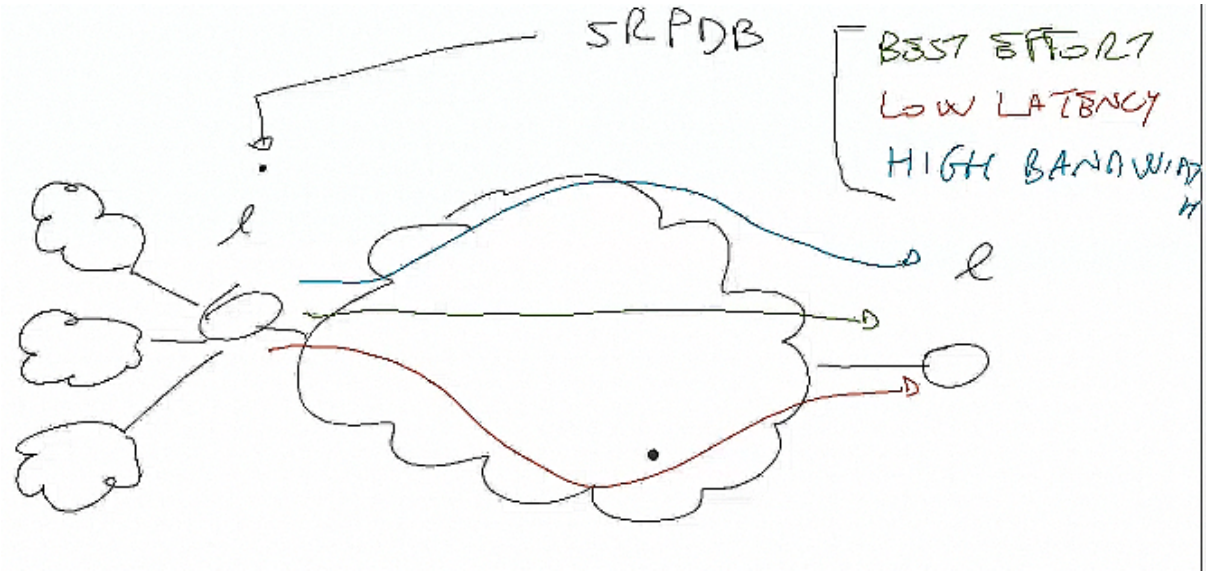
2. Centralized Control Plane. In which segments are allocated and signaled by an SR Controller that also computes the source-routed policies for the traffic. The centralized node has the task of computing the segment list, finding good paths and installing in nodes rules called SR policy that allow the packet to follow specific paths. The controller is a SDN (Software Defined Networking) CONTROLLA

3. Hybrid Control Plane: complements a base distributed control-plane with a centralized controller. Es. when the destination is outside the IGP domain the SR controller computes a source-routed policy on behalf of an IGP node. The distributed part of the control plane is used to distribute the SIDs information and the centralized part, a controller called SDN (software defined networking), that controls the network according to certain policies. CONTROLLA

SR Policies and Multiple SL:
An SR Policy is the logical instruction that an ingress node uses to decide what segment list to apply on an incoming packet. So the policies are applied in the edge nodes

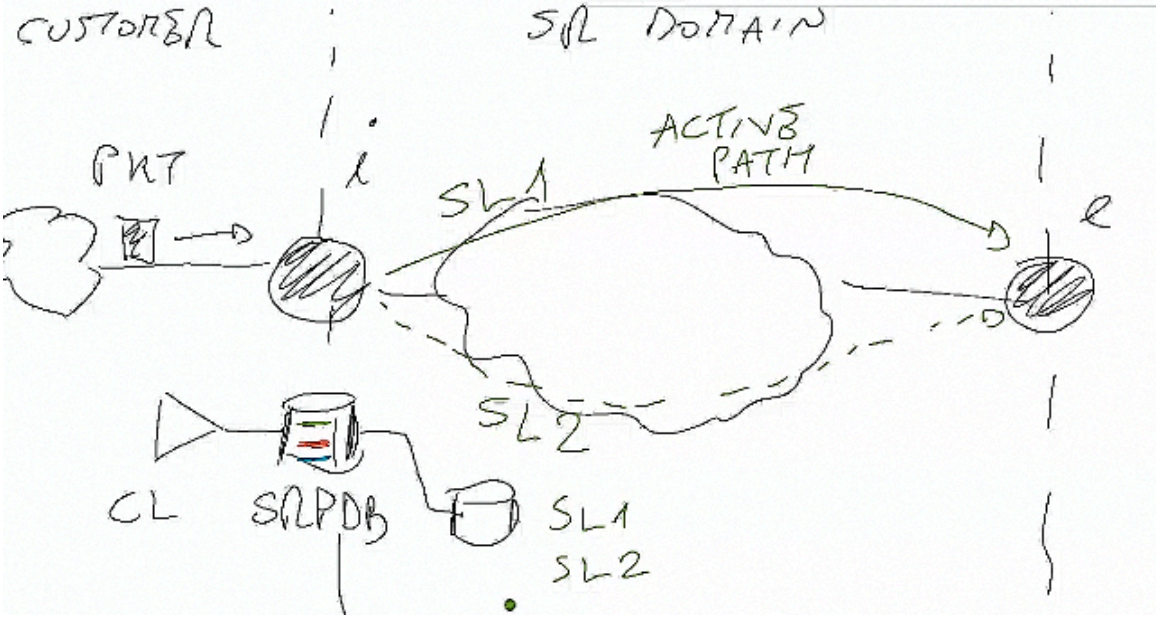


The policy, for instance, specifies the identity node, the identity of the egress node and the color, that associates the policy with an intent (e.g. low latency path in red). All the different paths (best-effort, low latency, high bandwidth…) which correspond to a color are in the SR Policy DataBase (SRPDB) of the ingress node i.

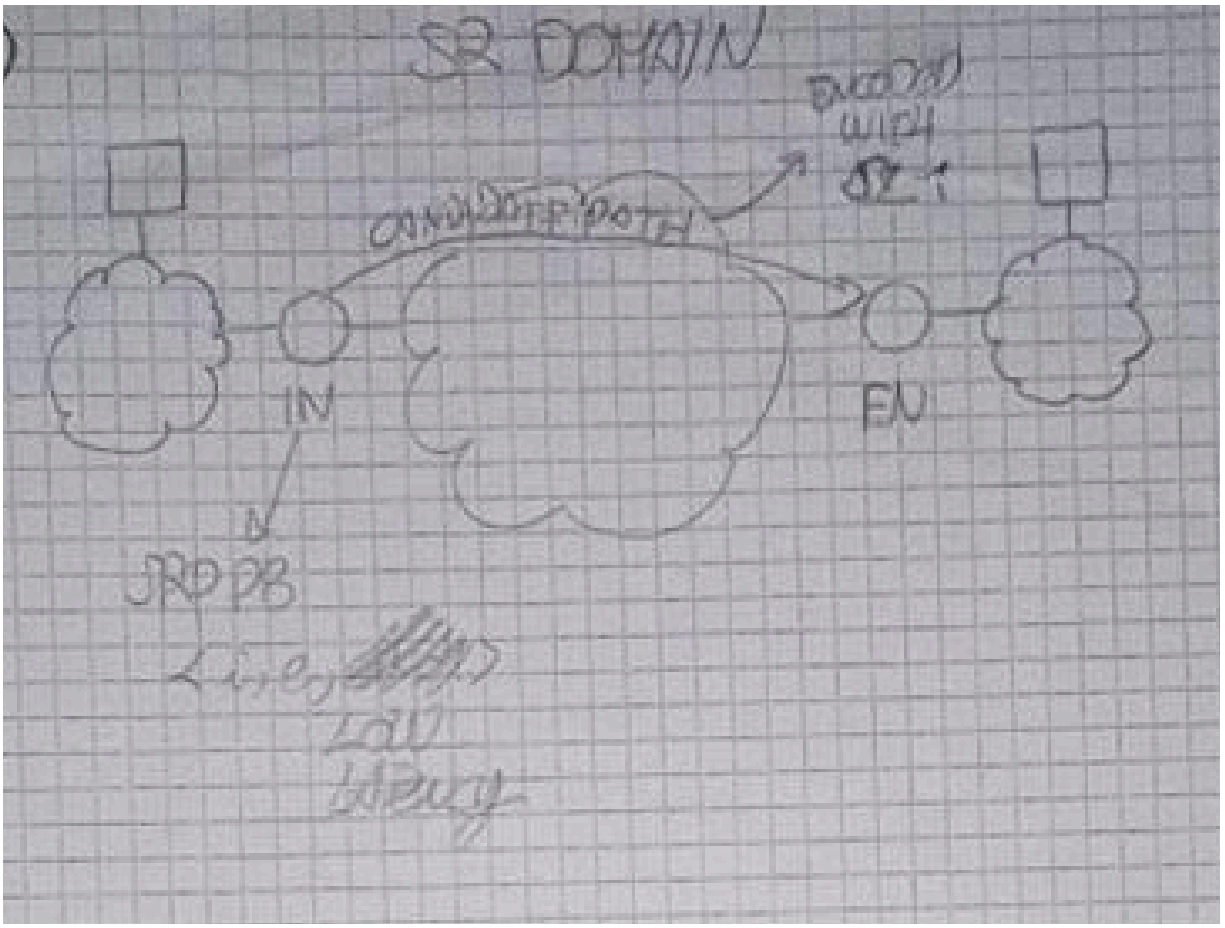


Let's assume the customer on top wants just connectivity. The ingress node has to decide the policy to apply. How to match the incoming customer packet with an entry in the SRPDB? We have to do classification: I have to look at the src ip, dst ip address, src and dst ports and the protocol field (UDP or TCP). So ok let's assume we decide we can steer the packet with the best-effort path. After that we apply the PUSH operation. Let's assume the

best-effort path. We can have multiple Segment-Lists SID-LIST: one is the working SL1 (active path), but there could also be a backup path for resilience with SL2.

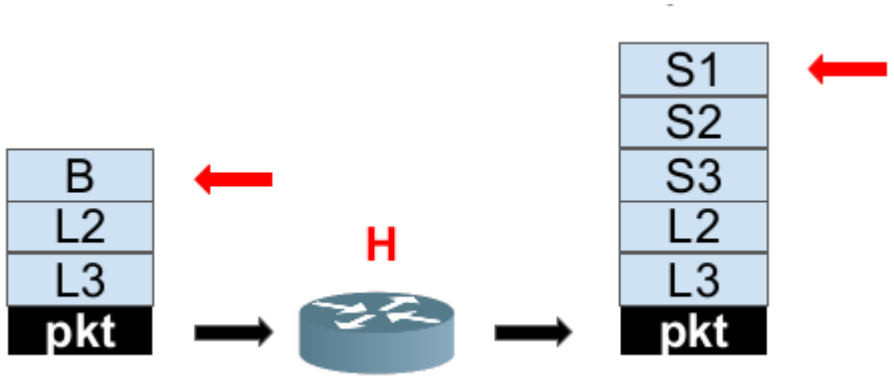


So there is another DB that tells us, for that specific color, what are the different SID-Lists available, namely what are the different candidate paths. One of them is the active one and, in case there is a problem with SL1 the ingress node can switch to the alternative SL.



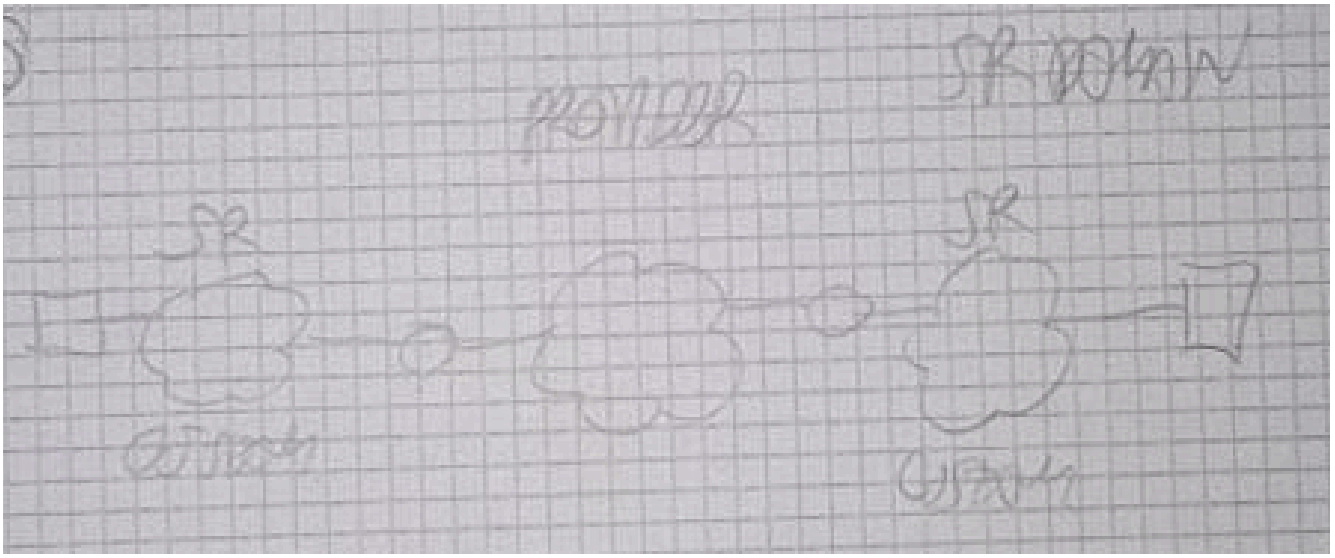
SL1, since it's the current active, will be pushed in each packet from the flow that wants to have low latency.

Binding SID:
I can interpret a policy <i, e, low latency> as a function. We can assign to this policy a SID, the binding SID. Imagine that the router H has a policy P, with SID B, that has low latency, for instance.



So I just push the SID B if I want to have low latency, because I want to be served with policy B when I arrive at H. Binding SID is a local instruction, because it is defined on the node H. The binding SID reduces the overhead since we reduce the segment list.

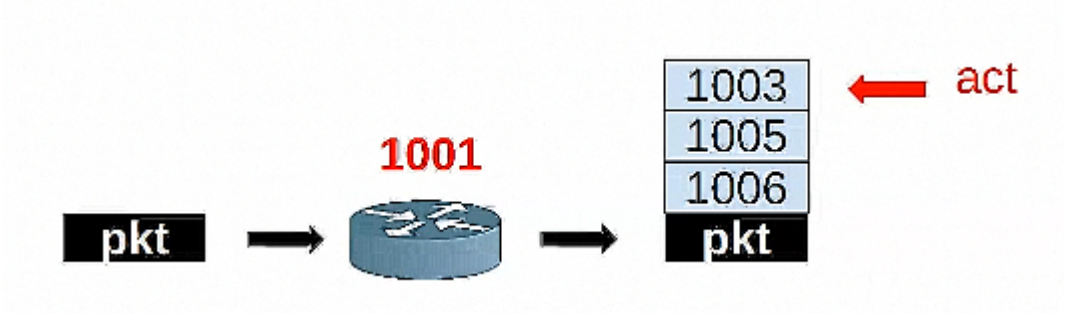
Let's consider this case:



The provider is using SR in his net, but also customers are, since they want to optimize the internal traffic. As long as the customers are in their own domain, they can decide the path they want. If the two customers are from the same organization the organization can decide on both edges. Let's assume we wish to pass through providers with low latency. I have to specify the binding SID B. This way the provider is not disclosing anything of its domain, the customer can't decide the path, just the service. The common place where binding SID is used is on border routers, but also intermediate routers can do that (we will see that after).

SR Forwarding (similarities with MPLS):

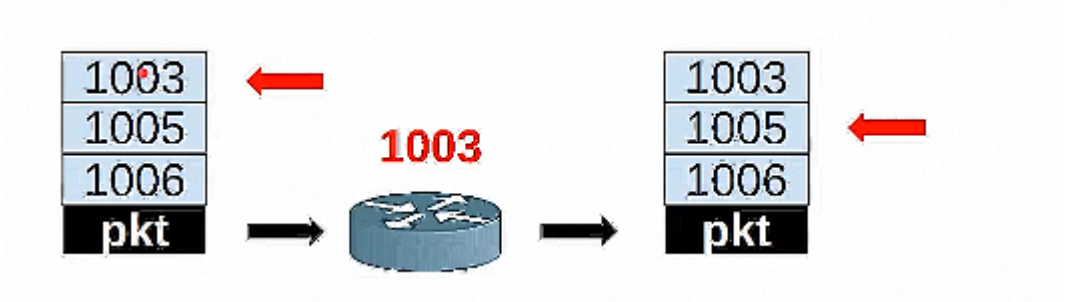
SR can be directly applied to MPLS architecture with no change when it comes to the forwarding plane. How a SR node processes the packet:



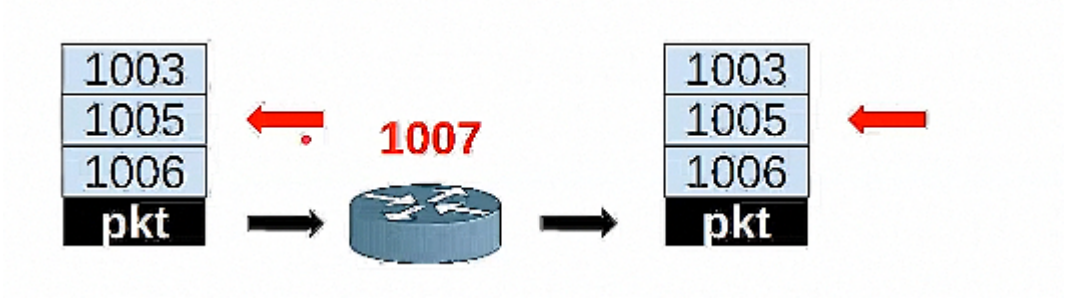
This is an ingress label, since the incoming packet has no segment list. We have to add a new header, so we have to do the PUSH operation, as in MPLS. Other similarities with MPLS:

a. 1003 is a SID as in MPLS we have the label
b. we have a segment list, as in MPLS we have the label stack

SR has also the NEXT (POP in MPLS) operation:



The segment has to be updated when 1003 is reached. So we move the pointer to the next instruction. If you remember the problem of troubleshooting the network in MPLS. Here I can just look at the packet with SR protocol and know all the list, because I didn't POPPED the instruction executed (as we did, indeed, in MPLS). Finally, the operation CONTINUE:



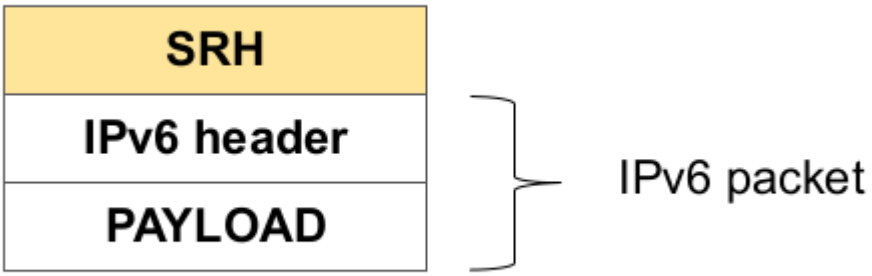
In this case I don't have to do anything, I just have to let the packet go. In MPLS we had to do a label SWAP from 1005 to 1005. Another similarity: when we apply a policy, the ingress node has to decide the policy to apply. How to match the incoming customer packet with an entry in the SRPDB? We have to do classification: I have to look at the src ip, dst ip address, src and dst ports and the protocol field (UDP or TCP). So ok let's assume we decide we can steer the packet with the best-effort path, I have to add a segment using the PUSH operation. So as you can see it is analogous to the MPLS and we don't have to create another infrastructure to replace MPLS with SR.

SR Data Plane:
If we define a new protocol do we have to update all the possible systems in the world? This is the problem called Internet Ossification. The process of updating the functionalities in a network is very complicated and slow. After all, the functionalities of SR are very similar to MPLS and IPv6. So as I mentioned before, we have SR over IPv6 (SRv6) and SR over MPLS (SR-MPLS). Since IPv6 it's extensible, for SR we just create a new header.

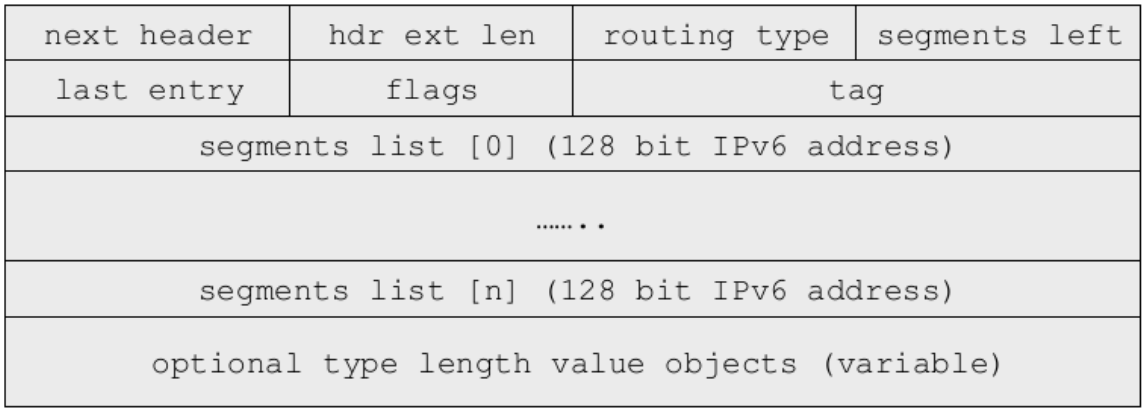
SRv6

Segment Routing extension Header (SRH):

SRH is defined as a routing type header.

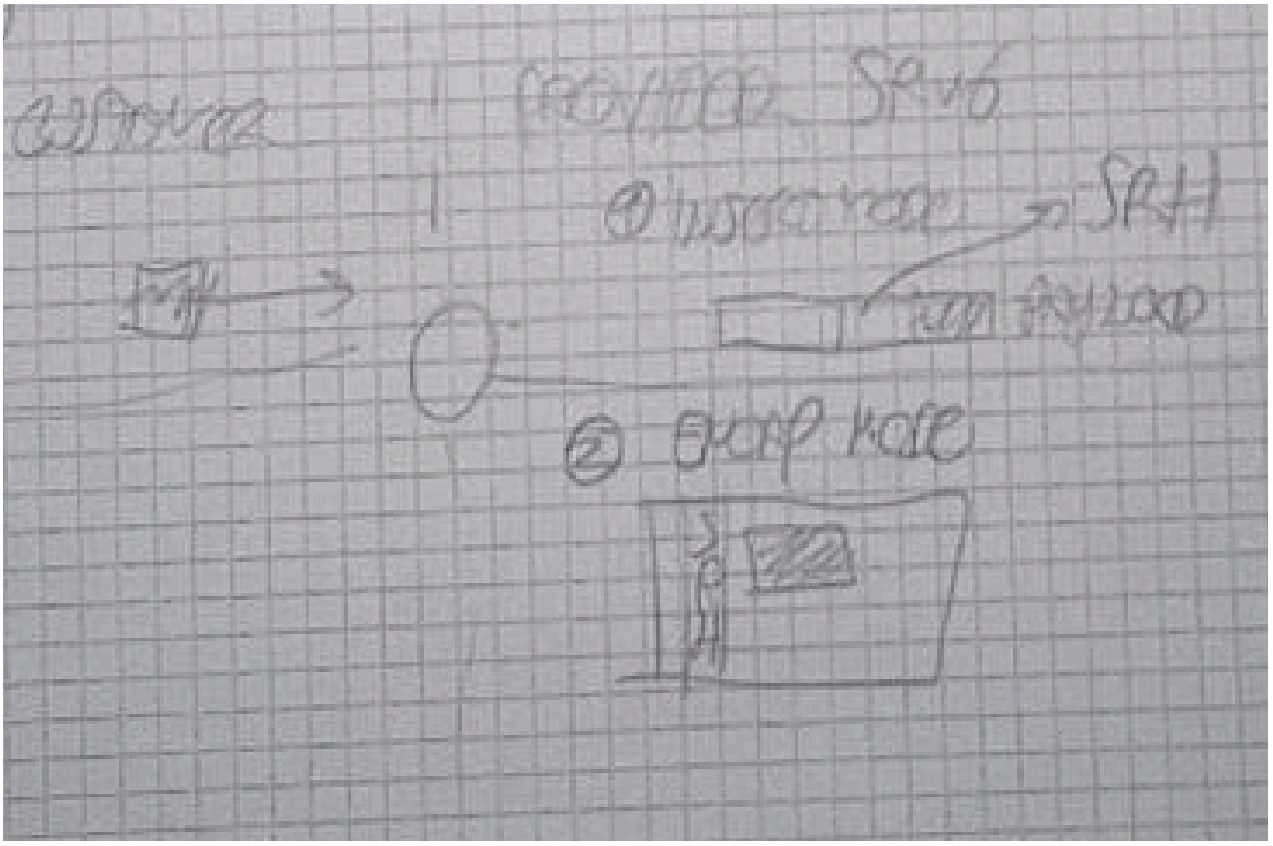


SR Header:



1. Next Header
2. Hdr Ext Len: length of the SRH header
3. Routing Type
4. Segments Left: contains the SL index of the next segment to inspect. So it's a pointer (the red arrows in the previous pictures) that is decremented by 1 at each segment inspection, until 0.
5. Last Entry: contains the SL index of the last element in the SL
6. Flags
7. Tag: to tag the packet as part of a certain class of packets that share the same properties.
8. Segment List [n], being each SID an IPv6 address. SL[0] is the LAST Segment (DA CONTROLLARE SE SL[0] È L'ULTIMO O IL PRIMO).
9. Type Length Value (TLV)

The SRH header is added at the node that originates the packet and at each ingress node of an SR domain. We can add this header in two ways, so to use it also in IPv4 domain:



How can we deliver SR messages through IP? We have two options:

1. Option 1 is to create an SRv6 packet INSERT MODE simply insert the segment routing header on the original packet. I can use the insert mode only if the customer is using IPv6
2. option 2 ENCAP MODE, this is the common mode. If the customer is an IPv4 I can use the encap mode, since the IPv4 packet is not going to be inspected in the provider. The encap mode could be useful also to create VPN.

L'USCITA È IPV6 PER ENTRAMBI: L'INSERT SI FA SE IL PACCHETTO DI DEFAULT È IPV6, MENTRE IN ENCAP MODE IL PACCHETTO PUÒ ESSERE ANCHE IPV4, E SE È IPV4 SERVE L'ENCAP MODE.

SHR Processing:

Only SRv6-capable routers whose address is in the Destination Address of the IPv6 packet can inspect the SRH. Let's make an example of SRv6 message passing. In red is the current instruction.

After the inspection, not only the pointer in the SRH Segment List is moved, but also the destination address of the IPv6 packet is changed with the next address in the SL.

An SRv6-capable node maintains a MyLocalSID Table that contains all the local SID and the corresponding instruction, or more generally, set of instructions (so a function). But why does this node have more than a single SID? A node connected to 5 interfaces has 6 SIDs, 5 for the interface + for the identifier + it could have policies and other services. Each policy and service has its own SID.

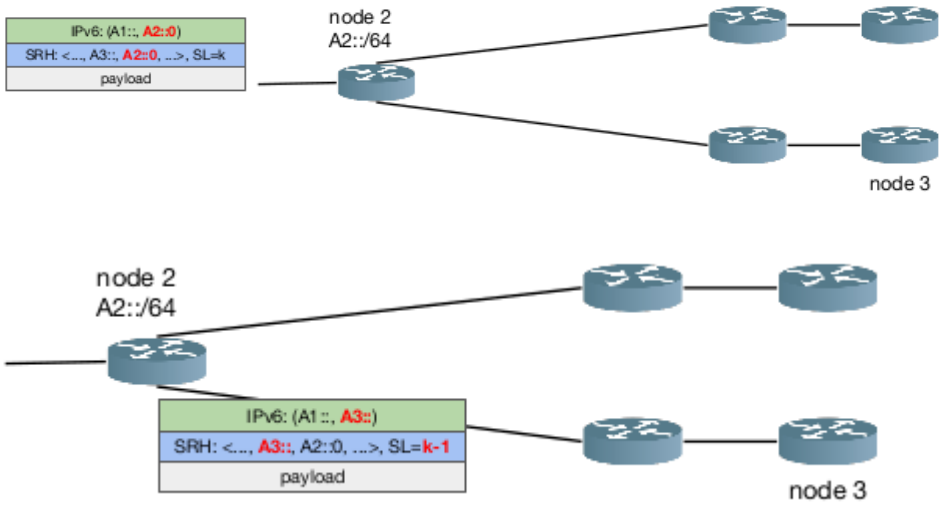
MyLocalSID Table

SID	instruction
...	...

Two basic SRv6 functions are:

1. The END function: updates the destination address with the next segment and forward the packet accordingly.
   Pseudocode of the END function executed in an SR Endpoint:

   IF SegmentsLeft > 0 THEN
   decrement SL
   update IPv6 Destination Address with SRH[SL]
   FIB[6] lookup on updated DA
   forward the packet accordingly
   ELSE
   drop the SRH
   now we have a simple IPv6 packet

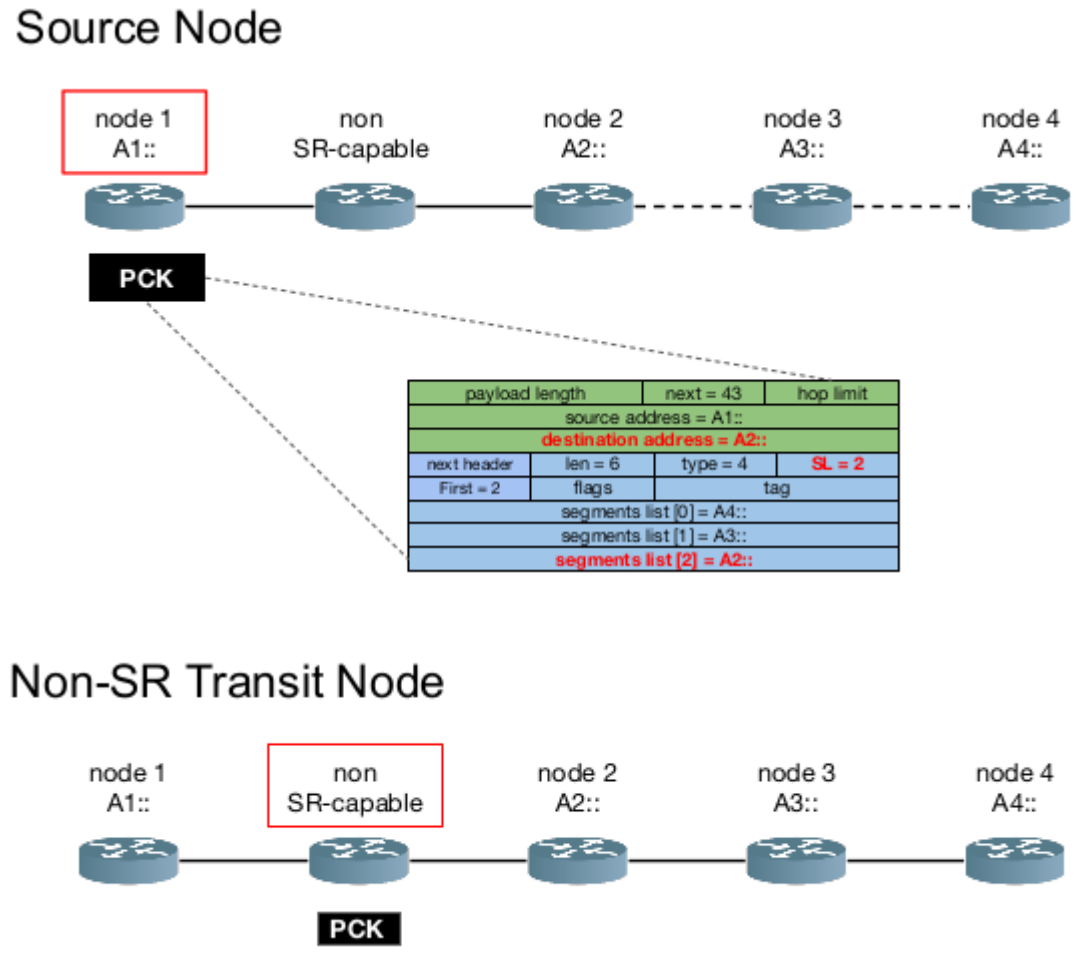


[6] the Forwarding Information Base, also known as Forwarding table or MAC Table.

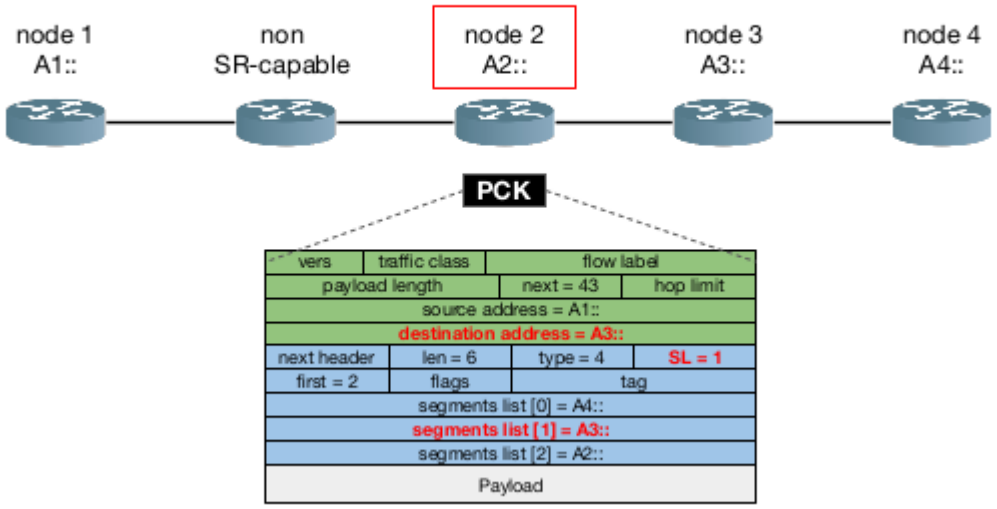
2. The END.X function: is associated with the adjacent, so send the packet through a specific link.

Non SR-capable router can't process the SRH, so it's transparent w.r.t. SR, they just do IPv6 forwarding (they don't read the blue part, just the green one).

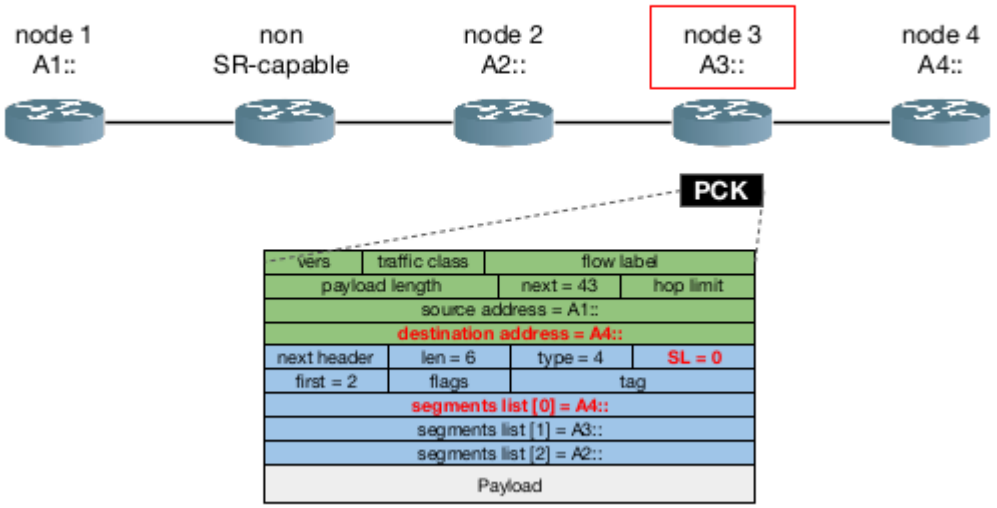
Let's make a complete example:



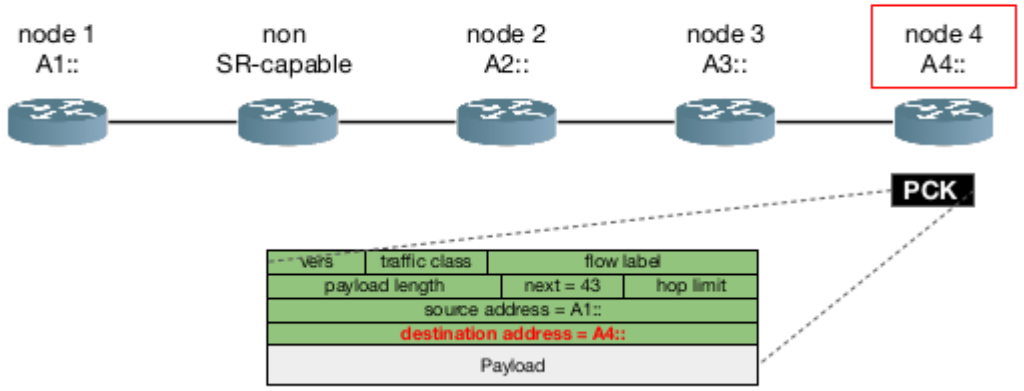
SR Segment Endpoints



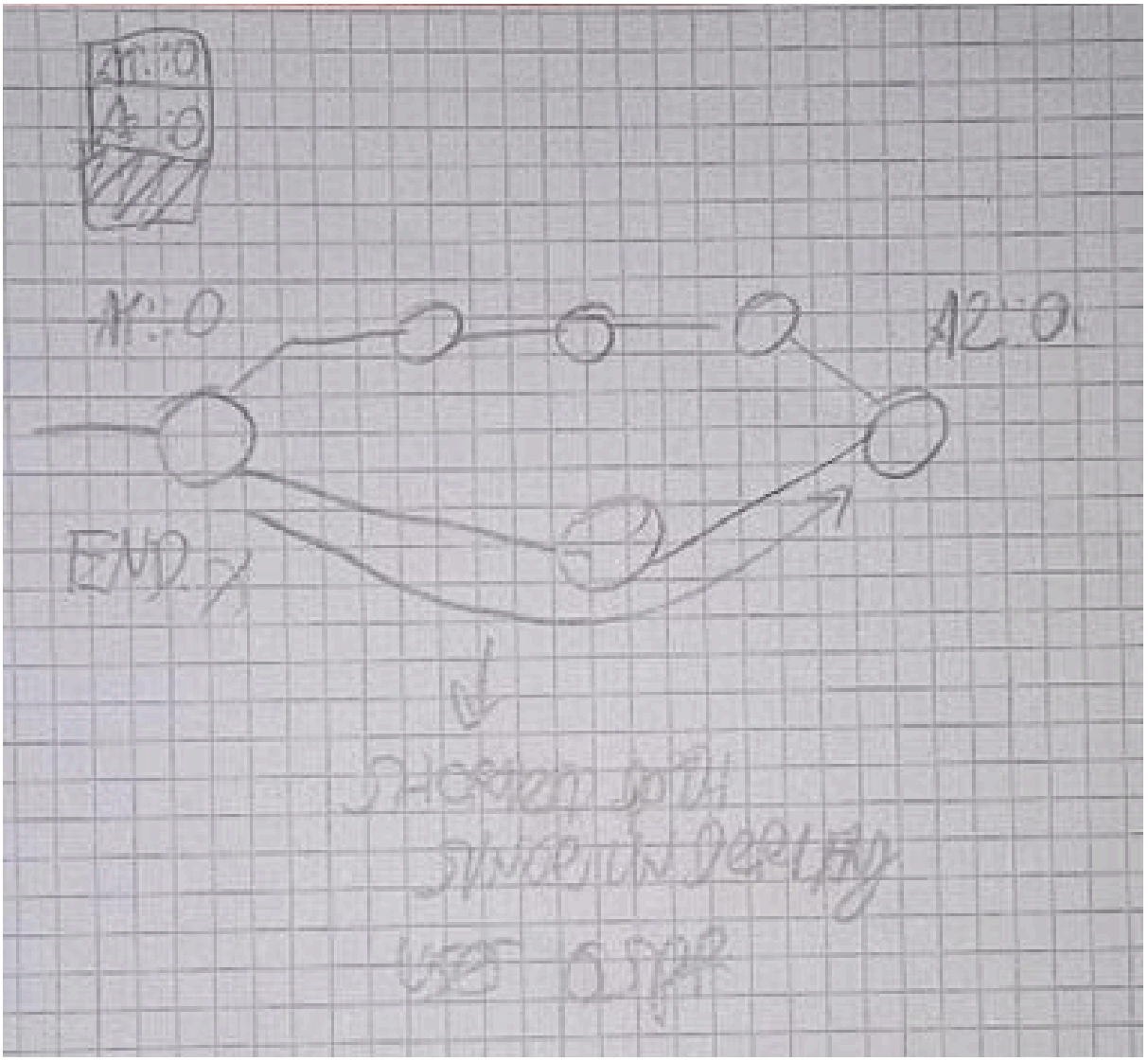
SR Segment Endpoints



SR Segment Endpoints



Let’s make an example with the cross-connect function:



what if we want to use the path above, but all the three routers above are non SR-capable. The cross connect comes in handy.

A1::0 MyLocalSid Table:

SID		BEHAVIOR
A1::0		END
A1::1		END.X (output interface 1)
A1::2		END.X (output interface 2)

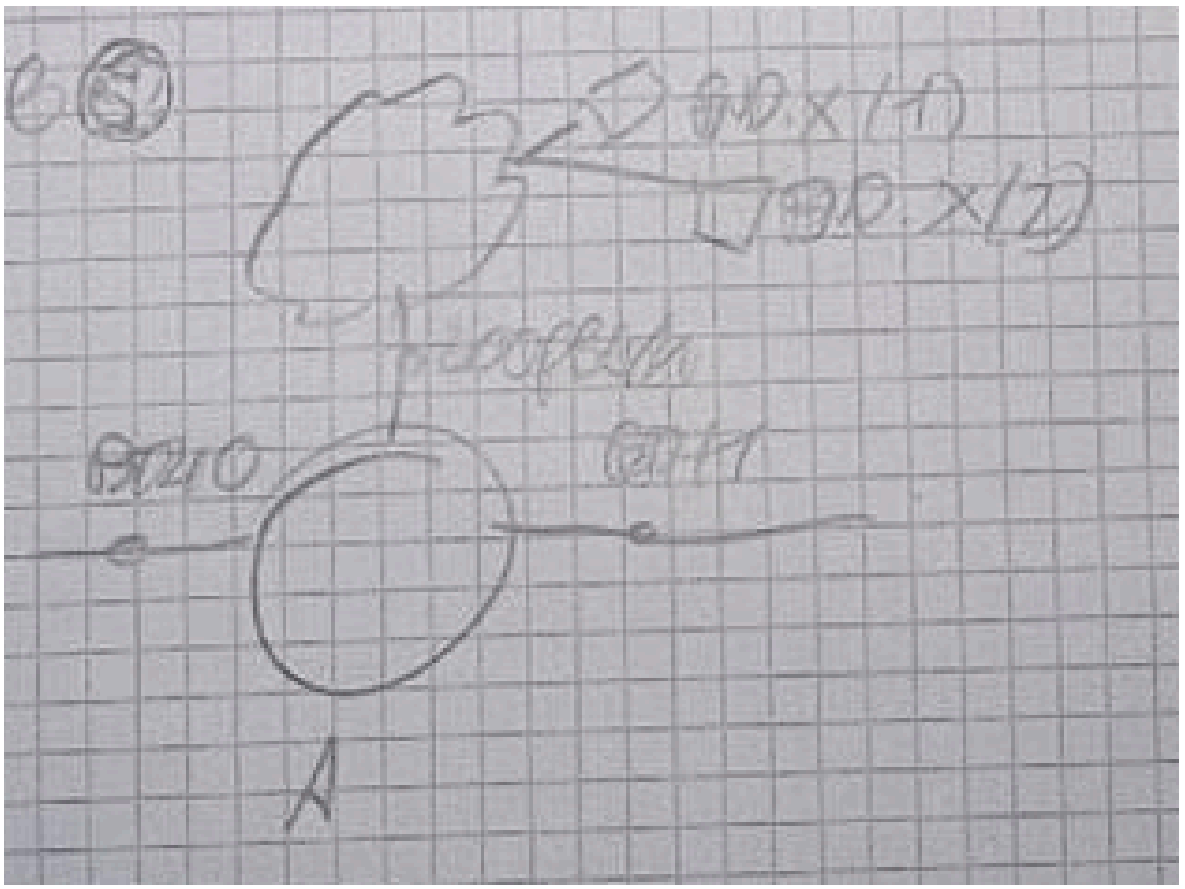
So I just need to use A1::2 instead of A1::0 in the header. After that, the non SR-capable will use their own shortest path (since they don’t inspect the SID) to deliver the packet, and so they will pass through our path and go to A2::0

SRv6 Segment Format and Network Programming

locator *function* *argument*

1111:2222:3333:4444:5555:6666:7777:8888

The SIDs we used have the same prefix, the *Locator*. The LOC part of the SID is routable and leads to the node which owns that SID.



We can think of that as if the loopback of the node is connected to a dummy network that doesn't exist, but has as prefix the locator, for instance A1::/64. All the functions available at this node are into A1::/64 block, so they have A1::/64 as prefix. All the functions are abstracted by hosts of this dummy network, so for instance the END function is A1::1, END.X (1) is A1::2, END.X (2) A1::3 and so on…

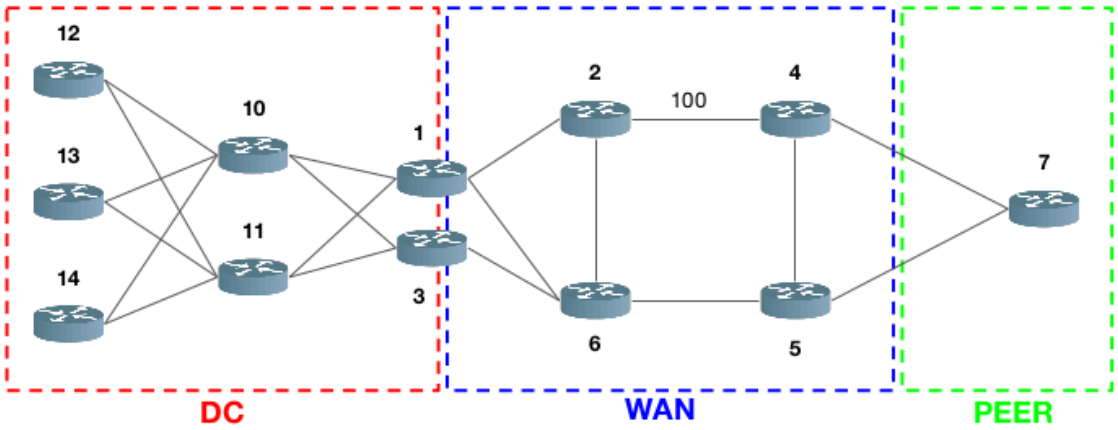
So the FUNCT part of the SID is an identifier of a *local* function. Since a function may require additional arguments, we can have a third section of the SID address, that is ARGS. So node LOC has a function FUNCT which needs arguments ARGS. FUNCT and ARGS can have what you prefer in terms of bits allocated, meanwhile the Locator is in general a /64.

This way, all the different SIDs of the node can be advertised by the other nodes in the network: so another node knows that if it wants to use a function that has LOC A1::/64 it has to come to node A.

I use the information of the locator to arrive to node A, once I am there I use the function part to specify the function I want (e.g. switching on the light) and then if there are more lights I specify that in argument (e.g. switch the light number 2). So if I specify a sequence of SIDs (segment list) I am writing a program. Since it's run on a network is network programming.

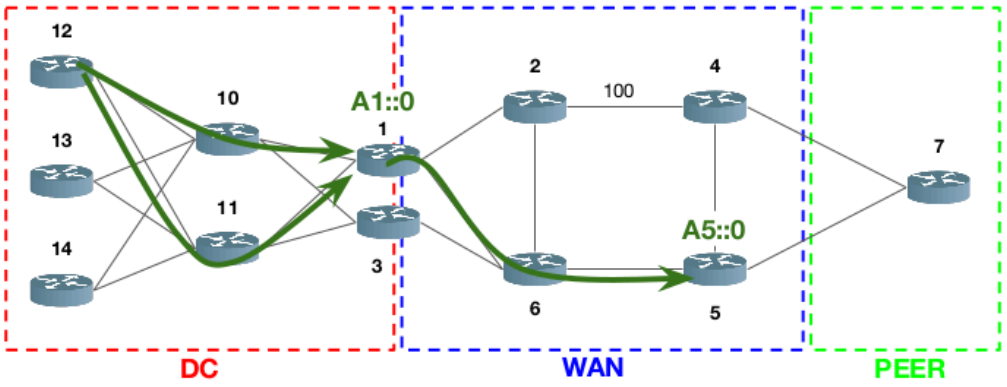
Each entry of a MyLocalSID Table indicated is in the form LOC:FUNCT:ARGS and indicates a function associated with the local SID. These function can be used to make whatever service you want, for instance:

1. Steer the traffic wherever we want. Let's consider this case:

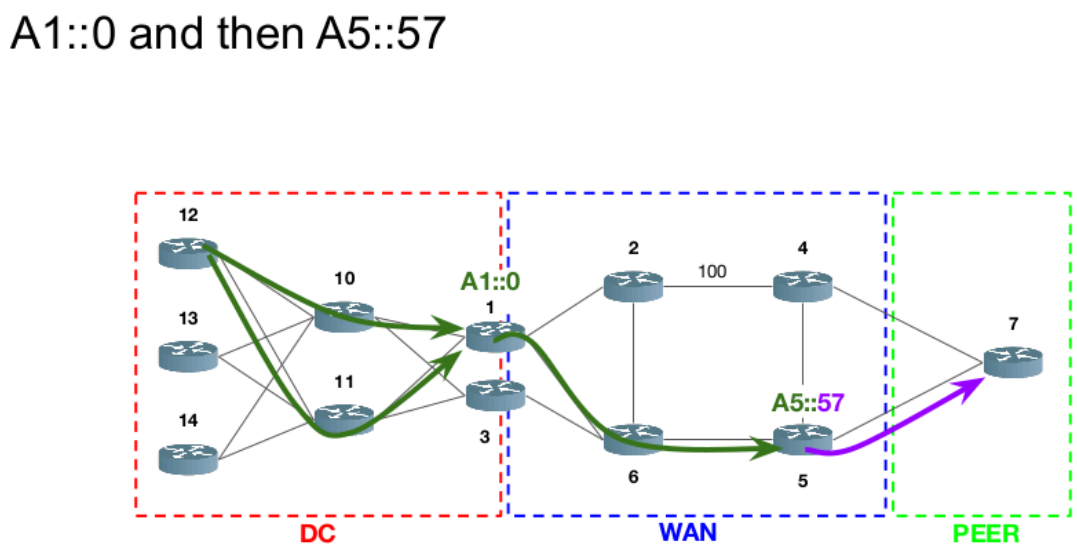


The datacenter is connected to the WAN. The number 100 in the WAN is the cost, the same concept as OSPF. AK::/64 is the locator of the node with SID K, so, for instance, node 10 has A10::/64. CJ is the cross connect, so the End.X function on link < node C - Node J >. Starting from node with SID 12:

A1::0 and then A5::0

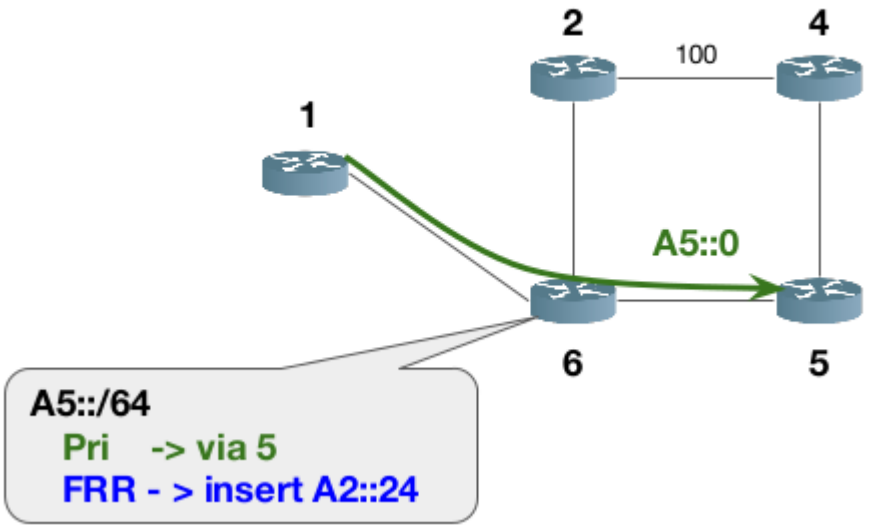


Note that SR performs load balancing, if the underlay provides equal cost multipath, SR uses all of them balancing the traffic.



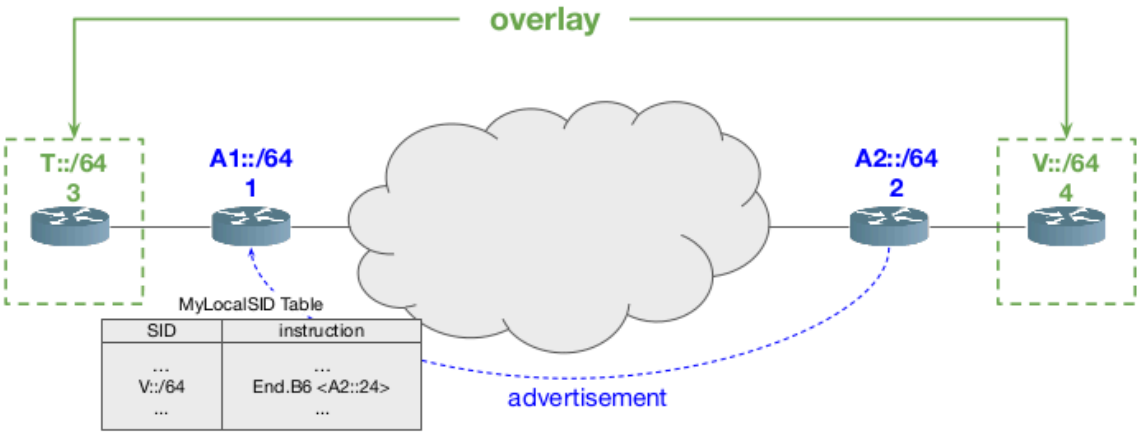
Here we are using cross connect to force the traffic to go in that link.

2. TILFA: Topology Independent Loop Free Alternate, that is a sort of fast rerouting:



We want to protect the traffic passing through 6-5 from failure. If the link fails FRR (Fast ReRouting?) inserts the other path, so it switches to the backup path applying a new SID on the incoming packet. I must use the cross connect since there is weight 100 so without cross connect I will go back to the failed link since in the underlay we use OSPF.

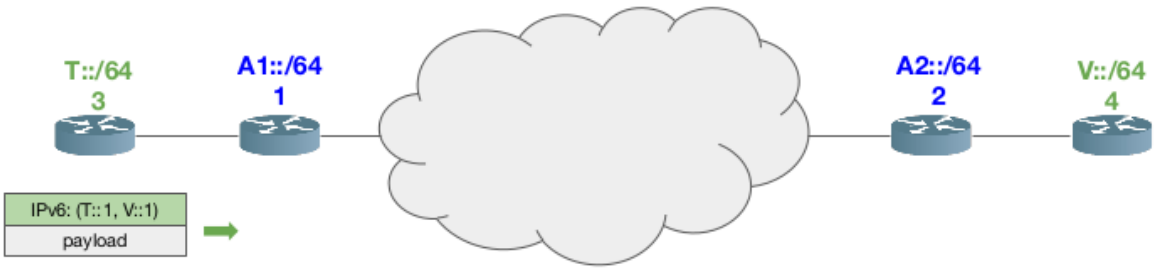
3. VPN: pass from CE to CE through PE without having to use multiple VRF. Let's assume a provider is offering connection to customers:



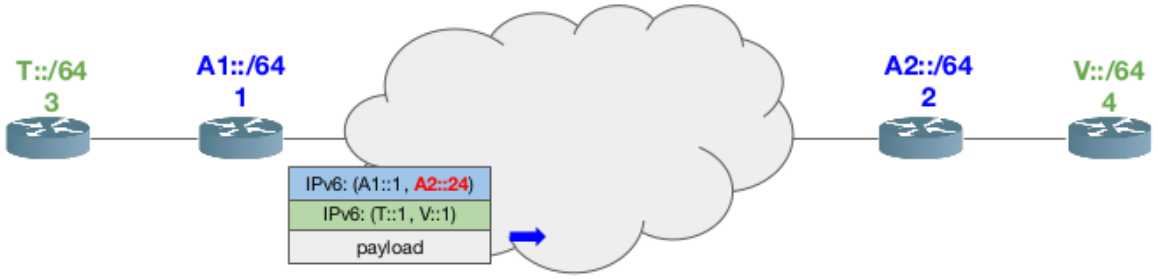
A1::/64 MyLocalSID Table is the one depicted in the image. If you want T to deliver messages to V we just have to specify the destination address with prefix V::/64 in the packet that T sends to A1; T, the customer doesn't have to specify anything else, since the rest of the net is owned by the provider and the complexity of the VPN must be in the PE.

Using this technique we don't need multiple VRF as in MPLS.. Then A1 will receive the packet and will look up in its MyLocalSID Table finding that, for addresses with prefixes V::/64, it has to perform function END.B6, so Binding SID that adds the following SL: <A2::24>, just one SID. VRF to separate the customer is not necessary anymore. Remember also that we don't have private addressing space in IPv6.

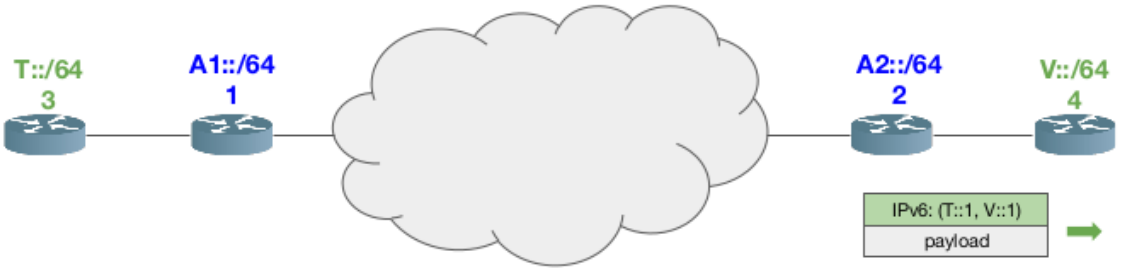
So, for instance:



T sends a packet with DA V::1/64, so it's in V::/64 prefix

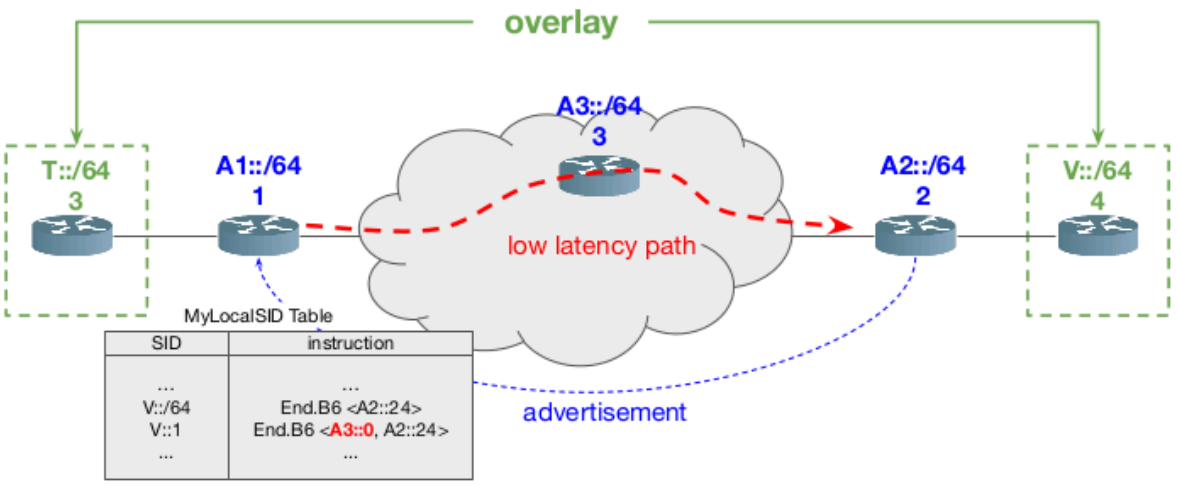


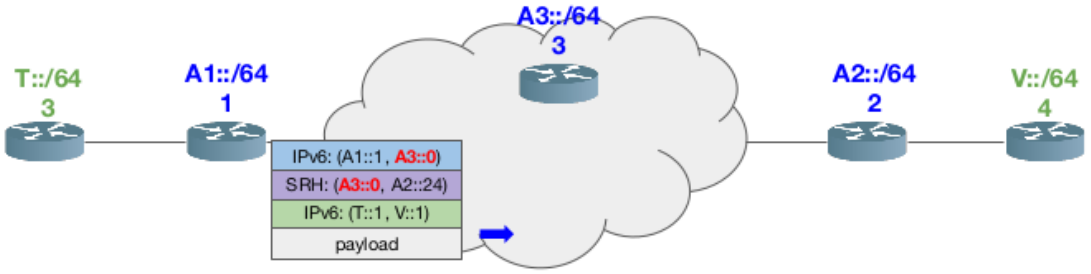
Once the packet is received by A1 the policy is applied and the segment is pushed. Note that we are using the encap mode, I encapsulated the IPv6 packet (in green) in another IPv6 packet (in blue). The blue one has the SRH but we don't see it because there is a single SID in the SL, but it's just a case.



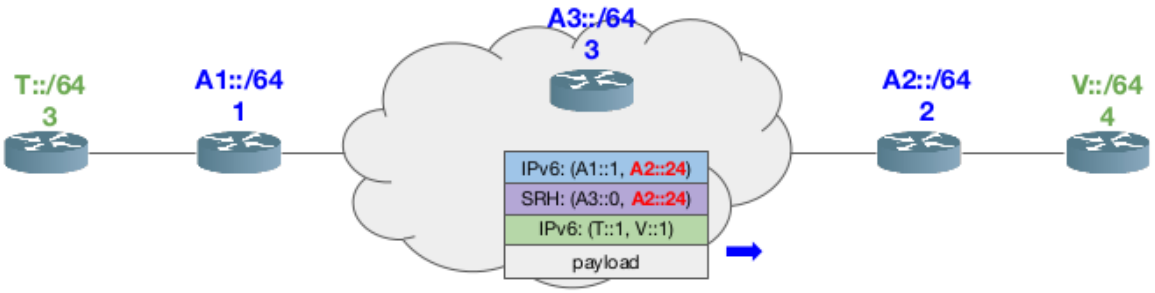
Eventually, the packet is decapsulated by A2 and sent to the destination.

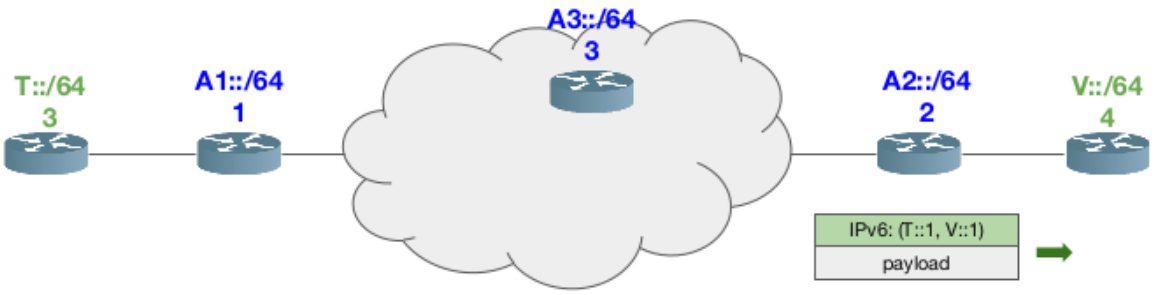
What if we want to add the possibility of using a path that has specific properties, such as low-latency:



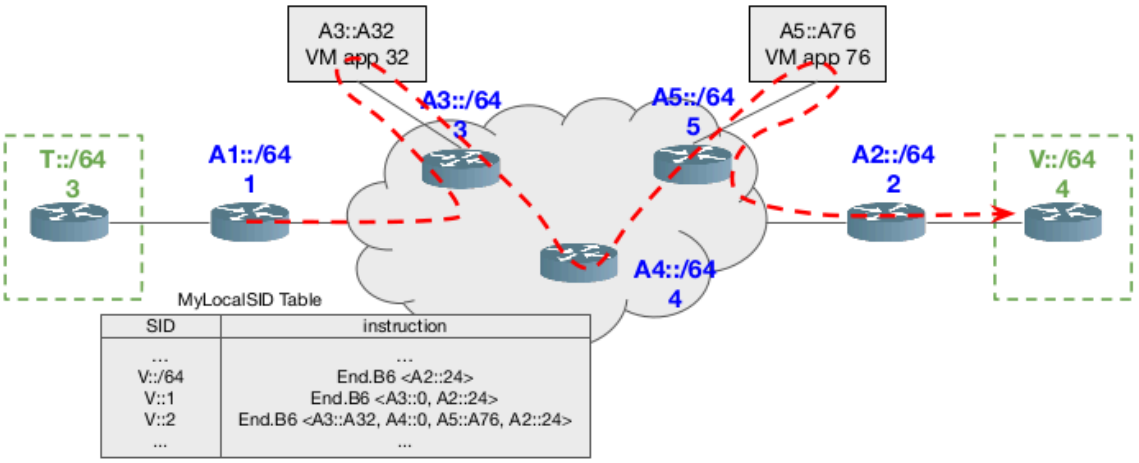


In this case we have the SRH since we have multiple SIDs.





We can do even more, adding the mandatory processing of our packets through some services:

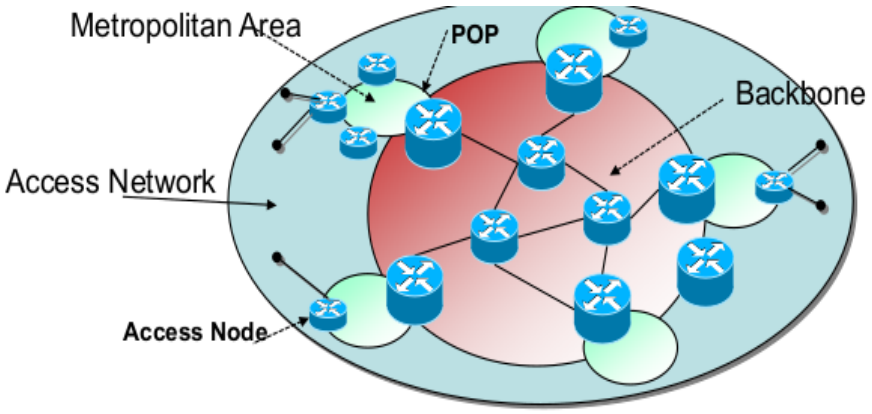


For instance, a VPN that is also secure needs to encrypt and decrypt packets. So we define two virtual functions that perform encryption and description. So when T inserts SID V::2 in its destination, the SL pushed by A1 is <A3::A32, A4::0, A5::A76. A2::24>.

So we have to visit A3, when you are there apply function A32 (encrypt), then we decide to visit A4 since it has a low-latency path for instance and then we pass through another function (decrypt), then we go to the destination as usual.

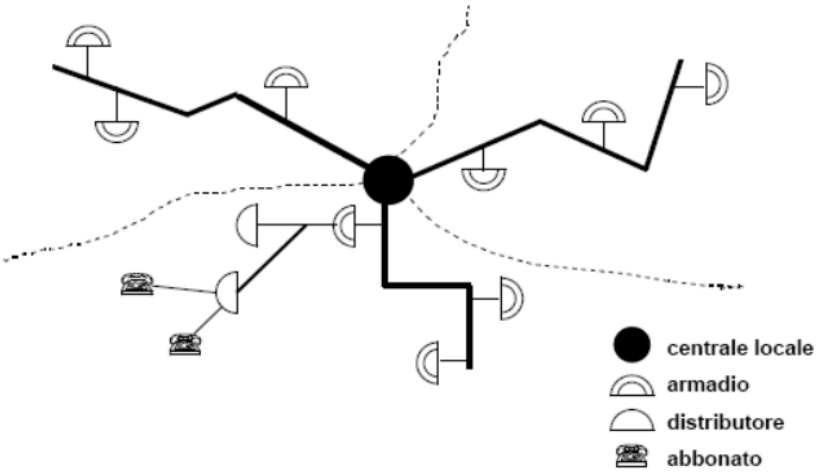
Introduction to Access Network

6/11, 13/11. 17/11, 20/11



In a communications network there are three part

1. Core part (the one studied until now), also called transport network
2. Peripheral Part:
    a. Metropolitan Part, in the middle of the peripheral and the Core
    b. Access Network, the very peripheral part. It connects subscribers to their immediate service provider. We can divide the access part in:
        i. Feeder Plant (or Feeder Network): from the service provider's central location to various local distribution points (e.g. armadi)
        ii. Distribution network (or drop plant): deals with the final connection from the local distribution points to the individual subscriber. So it manages the "last mile" connectivity.



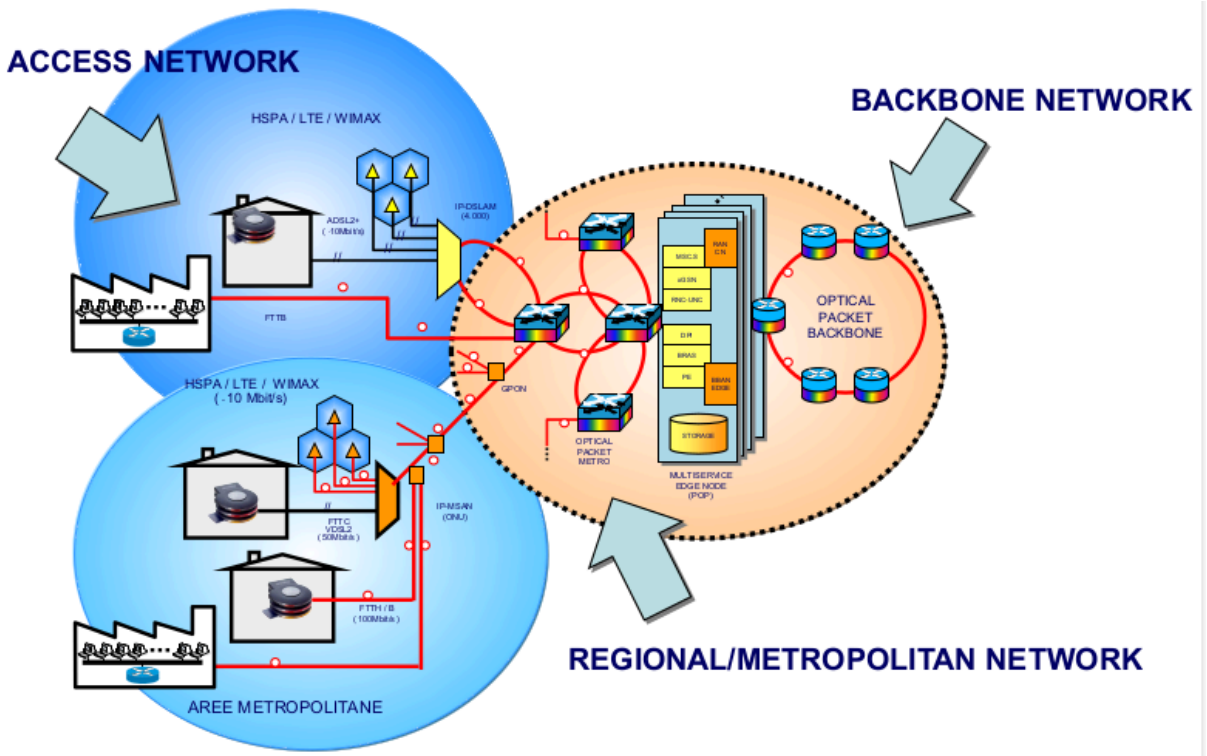
        iii. Edge network: in the middle between the core and the end-user. It is in the same position as the backhaul in the radio network (see cellular network part)

Comparison between Access and Core Network:

1. Media: while the core network is mostly an optical backbone, the access network is heterogeneous. Since historically carriers used a very capillary network of copper lines to offer phone service, nowadays the same line carrier broadband services such as DSL. Carriers are also investing in fiber optic, since it's high-speed and stable.

2. Topologies: while in the core there is the mesh topology, in the AN there is not mesh, but we have:
    a. Ring topology: a fiber with on top some routers. Very used in the metropolitan area. Rings are typically built up using 2 fibers, to have robust communication. Redundant infrastructure so if one fails you have the other.
    b. Star topology.

Let's see this example:

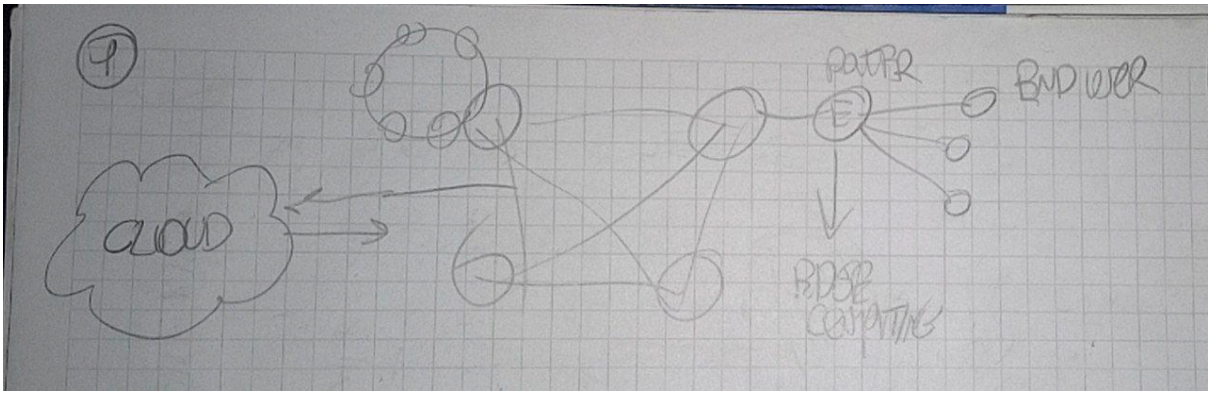


- The access network has a star topology. The multiplexer is the center of this star
- The regional metropolitan part is on the other hand done by using rings.
- The backbone can be a ring or a mesh topology

Edge Network:
The edge is the borderline. Nowadays the edge is also the part that implements the intelligence functions that are not performed in the core network. For instance if the core uses MPLS, an edge switch may perform the intelligent function of selecting the path through the network for a packet. So the edge is the smart part in this case, while the core is the “dumb” part that just switches the packet.

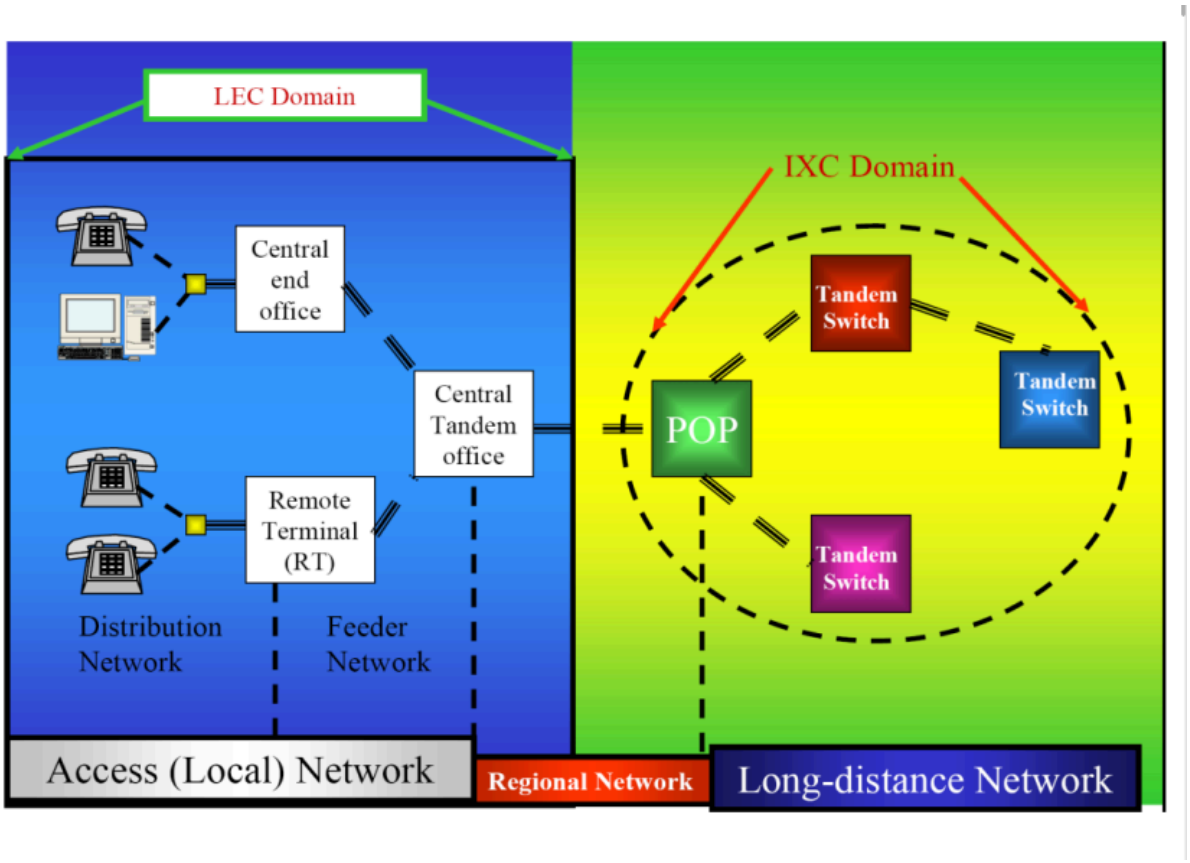
We can have the intelligent computing:

1. In the Cloud (so outside the network)
2. Edge computing, this is the trend today. We want to provide the intelligence close to the end user



Why is it better to provide the intelligence closer to the end user? For instance, when you have to authenticate to enter somewhere, this is done by the authentication server. The closer the authentication server is to the end user, the less latency there is. If the thing is urgent we want to put the authentication nearer to the end user.

Some Terminology:



- Exchange Area: we have an access LOCAL network and a LONG-DISTANCE network. In between the regional network.
- Local Exchange Carrier (LEC): company that provides services in a specific Access LOCAL NETWORK:
    - Incumbent LEC (ILEC): is the LEC that owns and manages the physical infrastructure in the area. Telecom is an ILEC in Italy.
    - Competitive LEC (CLEC): providers that are not the owner but use the infrastructure owned by the ILEC or own just a part of the infrastructure.
- Central Office (CO): a sort of switch where elements are interconnected. Physically these are big rooms where all the network devices are put. There is a Hierarchical architecture: we have Central Offices near the user (Central End User), and Central Offices higher in the hierarchy and distant from the user
- Trunks: cables for fiber optic.

- Local Access and Transport Area (LATA): a government-defined geographical boundaries in which government-defined phone companies can provide services
- Inter-Exchange Carrier (IXC). Carry inter-LATA traffic

Type of Access:
AN has two types of Access:

1. Wired Access: more reliable, stable and supports higher bitrate than wireless. Examples of wired access technologies:
    a. Fiber Optic (FTTx)
    b. Copper Based (xDSL)
    c. Cable Access[7]
    d. Powerline[8]
2. Wireless Access: more flexible and easier setup than wired. Moreover it supports mobility. We have terrestrial and satellite access. Examples of wireless access technologies:
    a. Fixed Wireless Access[9]
    b. Cellular Access

We will go in depth of FTTx, xDSL and Cellular Access.

[7] Used in USA TV. Is in copper but has a different w.r.t. to DSL

[8] Transfer data on electricity cables. Used in houses and outside, such as in railways, in which the network follows the train through the electricity cables.

[9] Also called Wireless Local Loop (WLL). An antenna fixed on top of a building (for instance) connecting with another antenna in the building of the end user. No mobility here. Simplest solution for wireless access

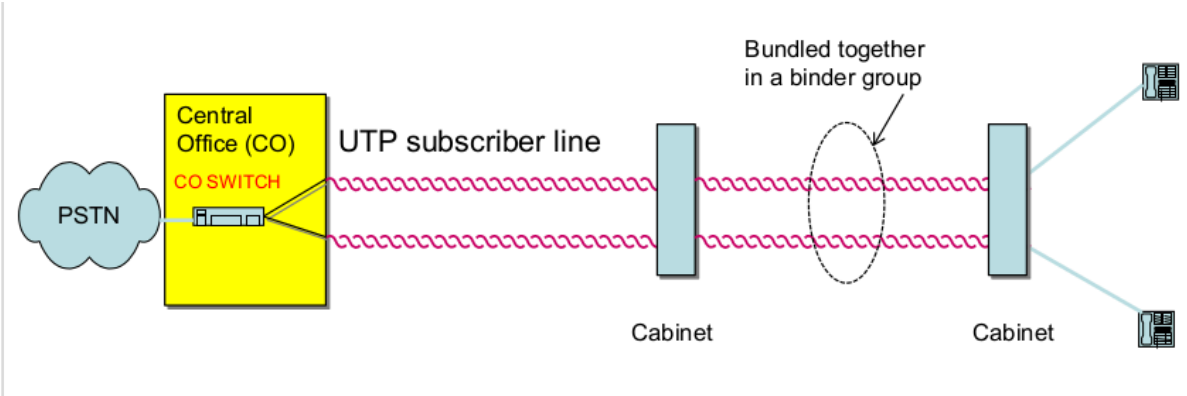
xDSL (Digital Subscriber Line)

13/11, 17/11, 20/11

xDSL is a family of copper based technology. xDSL leverages the widespread availability of the telephone network. But this creates a problem: the copper based access network was designed mainly for carrying POTS, so it's not optimized for transferring digital data. This is the log history of xDSL:

- ISDN Digital Subscriber Line (IDSL)
- High Data Rate Digital Subscriber Line (HDSL)
- Symmetric Digital Subscriber Line (SDSL), a standardized version of HDSL
- Asymmetric Digital Subscriber Line (ADSL), a version of DSL with a slower upload speed
- ADSL "lite" (or g.lite)
- Rate-Adaptive Digital Subscriber Line (RADSL)
- Very High Speed Digital Subscriber Line (VDSL)
- Very High Speed Digital Subscriber Line 2 (VDSL2), an improved version of VDSL

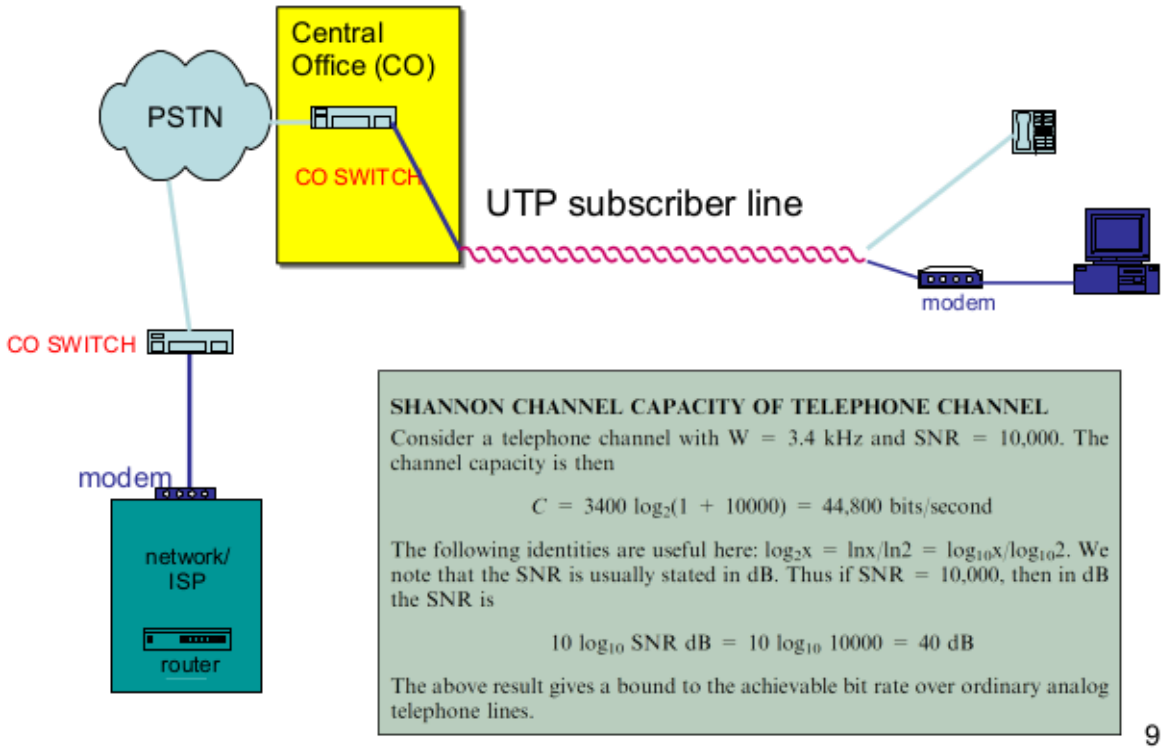
Pre-DSL Architecture:



Each end user had its own copper cable connecting them to the CO (centrale telefonica). So 1 end user 1 cable. Copper cables were bundled together in binder groups. Since in the cabinet (armadio) there is no switch, routing or conversion: the signal from the end user to

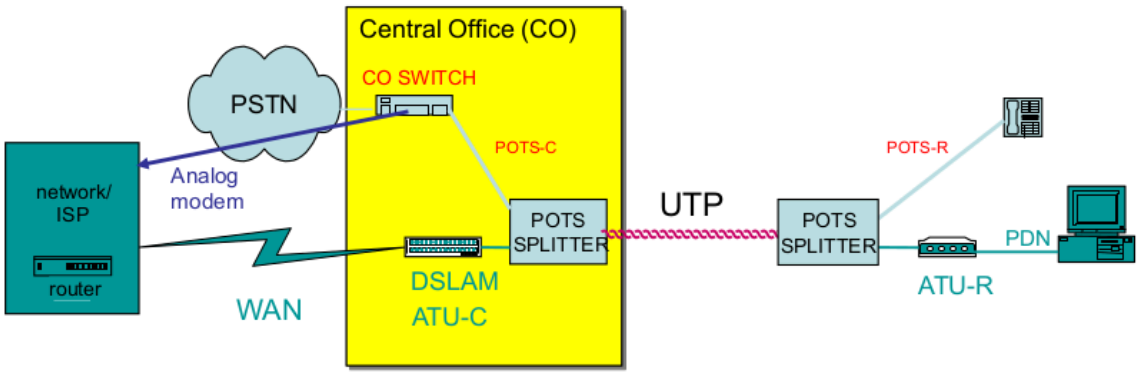
the CO and vice versa is always in the same copper cable. PSTN is Rete Telefonica Generale; it's like the "Internet" cloud in IP diagrams.

An evolution of that was the Analog (V.90) modems, that was the first try to transmit some digital signals with the analog telephone infrastructure.



The shannon channel capacity tells us that the maximum data rate for a digital signal in this architecture is 44,800 bit/sec = 44.8 kb/sec

Asymmetric DSL (ADSL) Reference Model:



As shown in the picture, we have a splitter and a modem per line at both sides. To use the same copper as the telephone we provide a device able to modulate and demodulate the signal with Water Filling. This device is called ADSL Modem, or ADSL Termination Unit. If I have a modem in my house (ATU-Remote) there is a twin model in the CO (ATU-Central) to demodulate what I modulated in my home. So a termination unit is present in my house and in the central office. ATU-C and ATU-R can perform both:

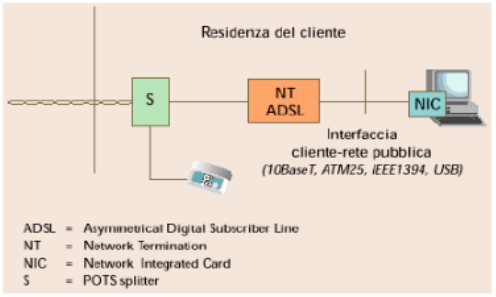
1. Frequency allocation (POTS, upstream, downstream)
2. Echo cancellation: more bandwidth than FDM (Frequency Allocation) but more complex

A DSL Access Multiplexer (DSLAM) houses a set of ATU-C interfaces, and serves as a central management platform. The Central Office or the Cabinet around the city should have the physical space to put the demux but in the cabinet around the city the space is very few so it’s more critical the topic of the physical space to find.

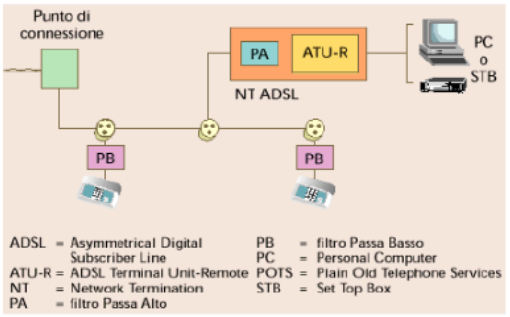
The splitter has to separate the part of the signal dedicated to the telephone (POTS) from the part that is for ADSL, using a multiband filter. The modem also integrates the splitter.

Splitter

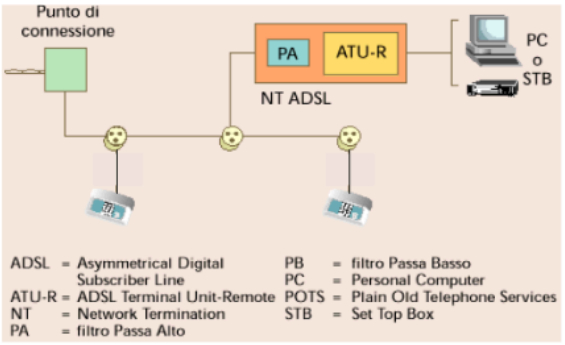
- Splittered



- Distributed splitters



- Splitterless (g.lite)



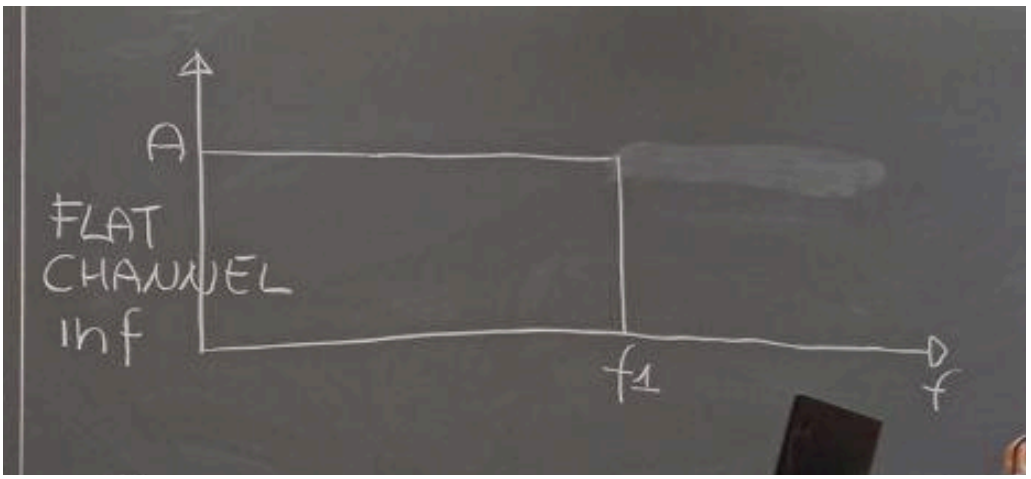
47

After passing through the modem, the ADSL is deployed in the house with Wi-Fi, Ethernet or Powerline. The splitter is typically centralized, but there were in the past solution in which the splitter was distributed and at the very beginning also a splitterless solution

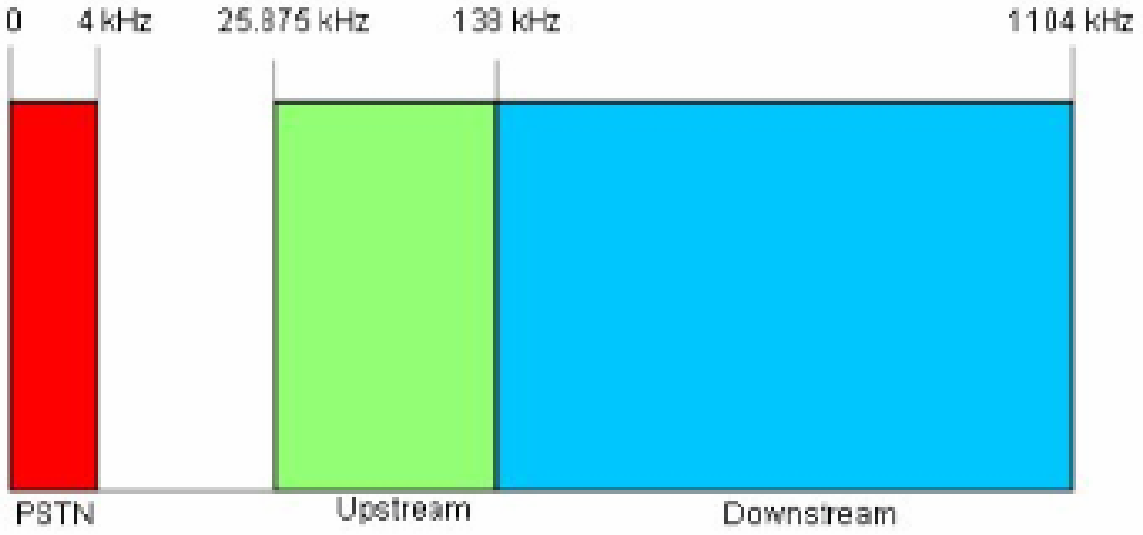
ADSL Frequency Allocation:

ADSL is Asymmetric DSL: we use more downstream than upstream since we are, as end users, mostly downloading things and navigating rather than uploading things on the internet. So we need two separate frequency bands, upstream and downstream. Upstream: end-user -> CO, Downstream: CO -> end-user.

Assume the channel is flat (we will remove this assumption later), and is a low-pass filter with f1 = 1104 kHz



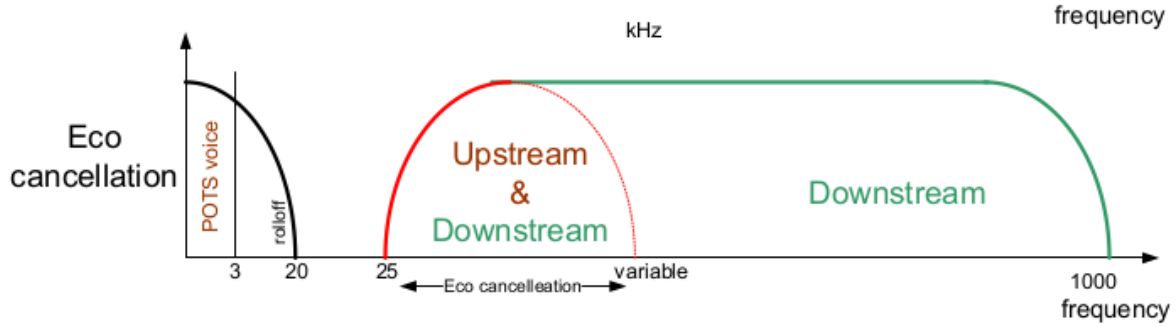
Thus, the frequency bands of ADSL is the following:



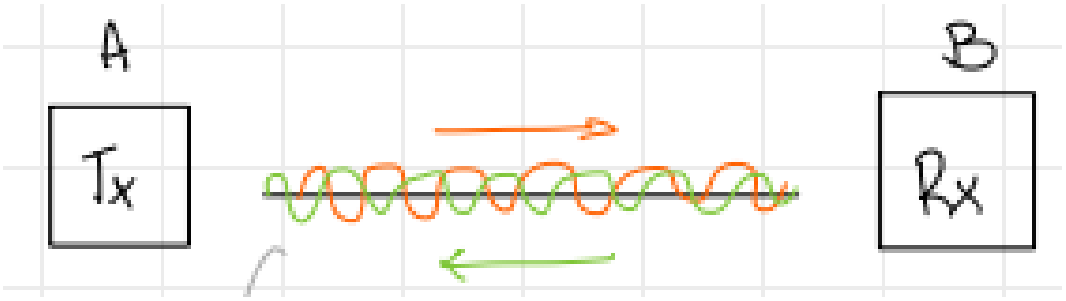
As said before, bandwidth [0, 4hz] is dedicated to telephones. Since the bit-rate formula is: C = B * log_2(1 + SNR[10]), a short bandwidth in frequency implies lower bitrate, so PSTN has low bitrate. The frequencies until f1 can be used to allocate frequencies to upstream and downstream. That was the frequency division approach.

[10] SNR is how much the signal (the good signal we would like to have) is higher w.r.t the noise that is added to the channel. SNR = power of the signal / power of the noise

There is another approach: *echo cancellation*. This approach allows us to get more bandwidth downstream by leveraging something that was used in the telephone.



In telephone (POTS) there were two directions not differentiated in frequency: when you transmit on the same frequency at the same time the interference is solved by canceling the echo. Let's do an example:
10101 <----> Central Office



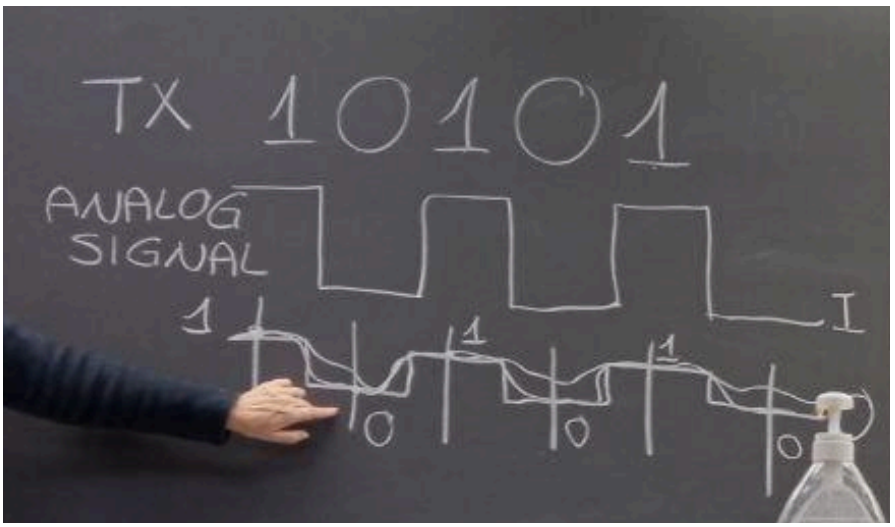
A and B are both transmitting and receiving at the same time and frequency. What a node can do is to cancel from what it is receiving what it is transmitting, since it knows well what it is transmitting. The viceversa is not feasible since a node doesn't know what it is receiving exactly.

ADSL Modulations

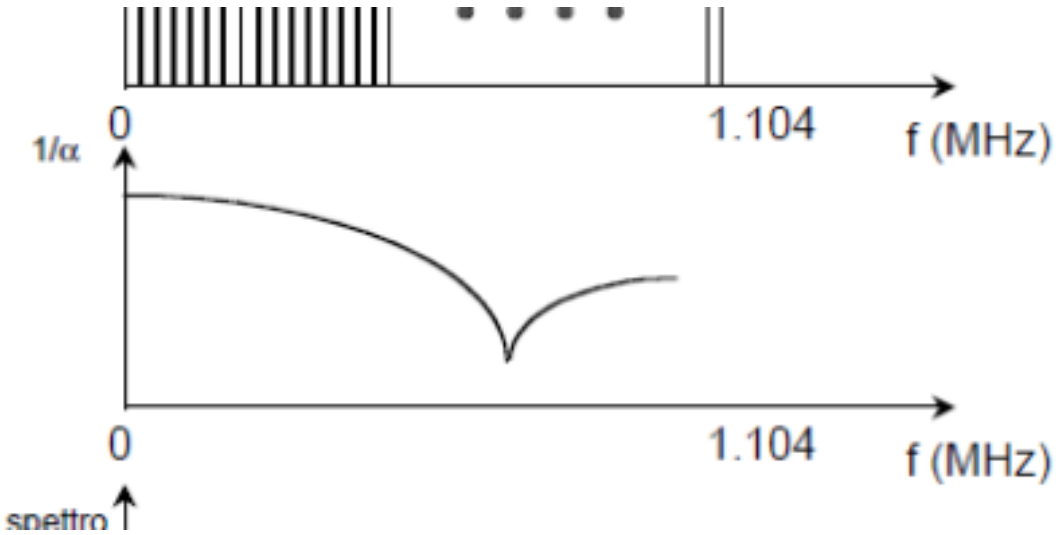
Two types of modulation:

- CAP: not treated, old method not strong as DMT
- Discrete Multi-Tone (DMT)

We will focus on DMT

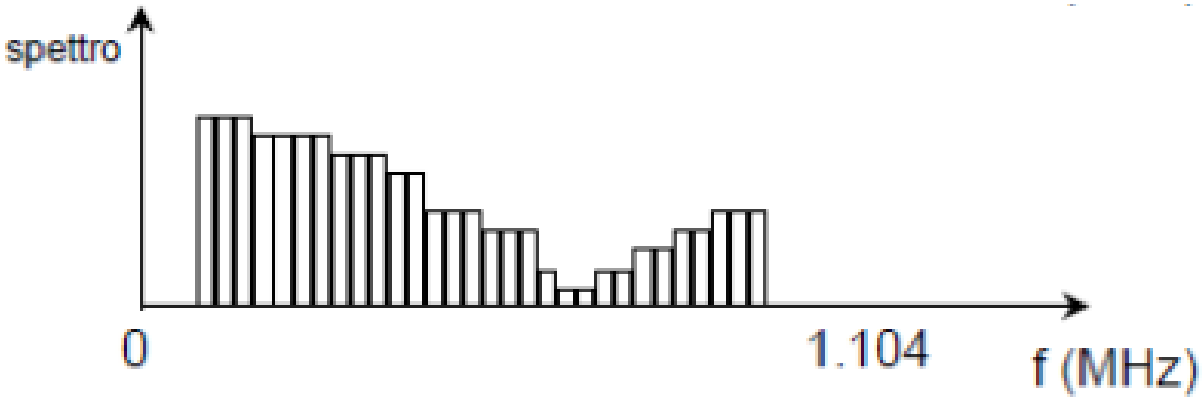


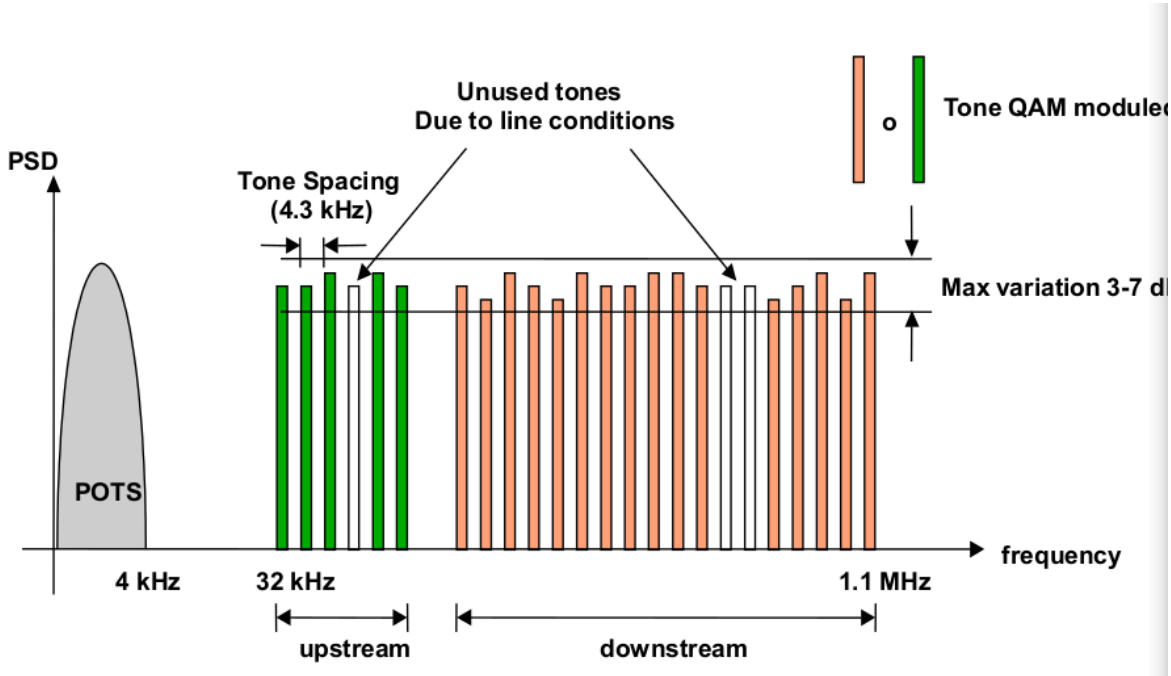
To transmit a digital signal (e.g. 10101) we use a shape that is a square wave. This is an analog signal, although it's discrete. Since the signal has to pass through a medium, and it will be attenuated, we would like to have a constant attenuation in frequency so the signal will not be distorted. This is called a *flat channel* in frequency. Being constant, the square wave will remain a square wave but attenuated. Unfortunately, a flat channel is ideal. In reality not all frequencies will be attenuated the same way. As we already know we have a low-pass filter with finite f1, the more frequencies are cut the more the shape changes becoming more smooth and less similar to a square wave, so we want f1 to be as high as possible. Moreover, the real behavior of the channel in [0, f1] is not really flat, but is something like this:



We can still transmit with xDSL, splitting in tones the channel, Instead of considering the whole [0,f1] channel together. Each tone corresponds to a specific frequency (f1.1 f1.2 f1.3)

and has 4 Khz bandwidth. In each tone the behavior is almost flat





We have 25 tones in the upstream and 249 in the downstream. Since the tone have different height, they have different attenuation, precisely:

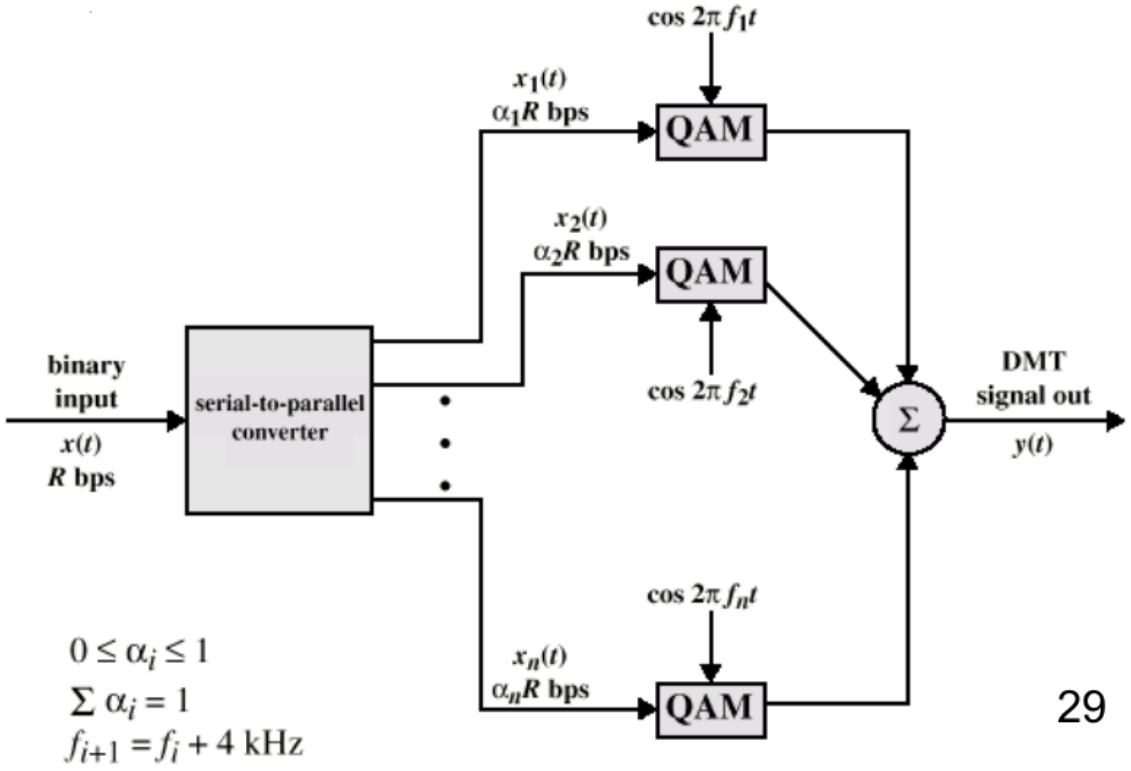
$C = B * \log_2(1 + SNR)$

If SNR is equal for each tone, the tones are identical only if they have the same attenuation, so if the channel is flat. But if the channel is flat there is no reason for dividing it in tones. So since the channel is not flat we have different heights and so different attenuation

As you can see from the picture above, there are some tones (the white ones) in which the attenuation is too high due to the specifics of the line so they can't transmit anything.

So if I have, let's say, 10101 that arrives in upstream at a given bit-rate, the signal it's splitted in 25 branches (# tones in upstream). We want higher bit-rate in tones that have higher height, so the tone with the best behavior (higher ones) has the higher number of bits sent.

So we have different QAM modulation tuned to have more or less bits w.r.t. the behavior of the tone,



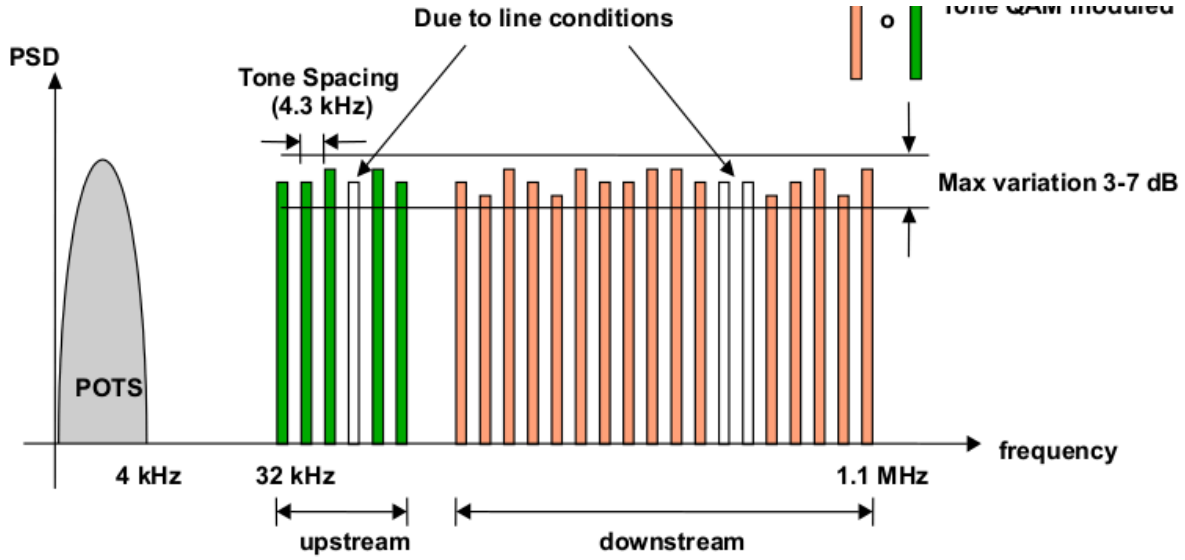
With DMT we get 256 sub-bands. Each sub-band is QAM64 or QPSK modulated, so it means that in each transmission I can adapt my communication since this thing is tunable. A tone is associated with a sub-band.

DMT theoretical maximum:

- Theoretical maximum upstream bandwidth:
  - 25 channels X 15 bit/s/Hz/channel X 4 KHz= 1.5 Mbit/s
- Theoretical maximum downstream bandwidth:
  - 249 channels X 15 bit/s/Hz/channel X 4 KHz= 14.9 Mbit/s

15 bit/s/hz is how much I can put in a tone. This is the nominal bit-rate but since each channel has its specific behavior, each ADSL user has a different behavior received since the channel conditions vary.

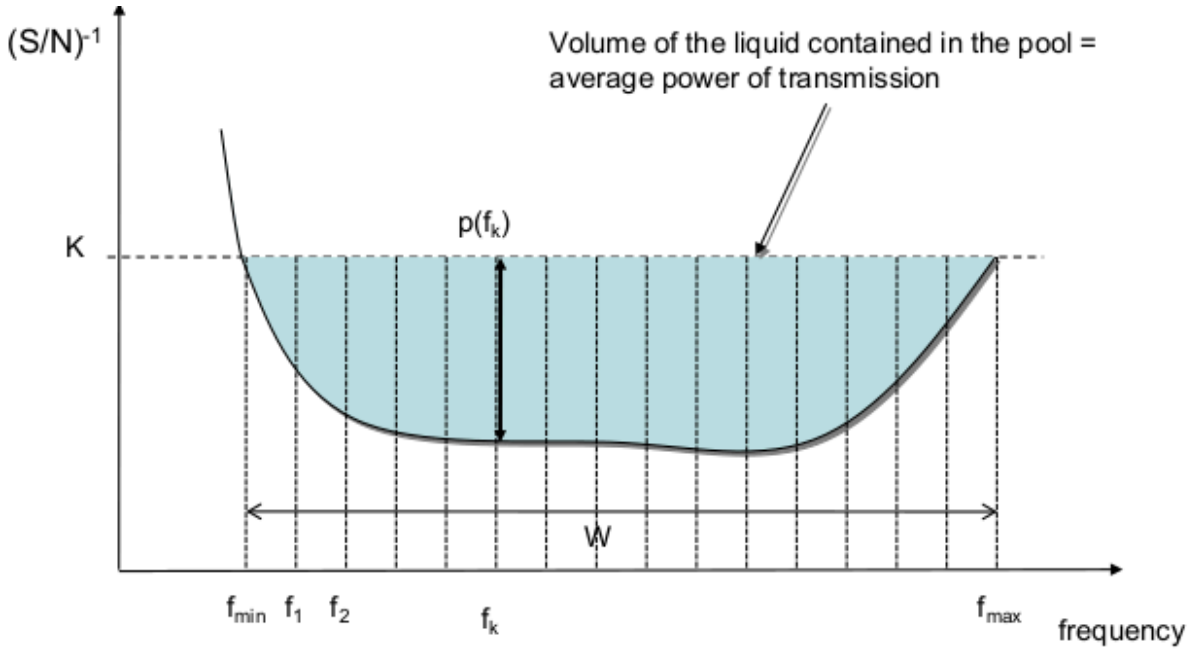
Since there is an overhead in splitting this, do DMT only if we need it, if we are sure that the channel is completely flat, is not useful to do that.



31

Moreover, we create *spacing* that is wasted bandwidth, so if the channel is flat it is counterproductive to do that.

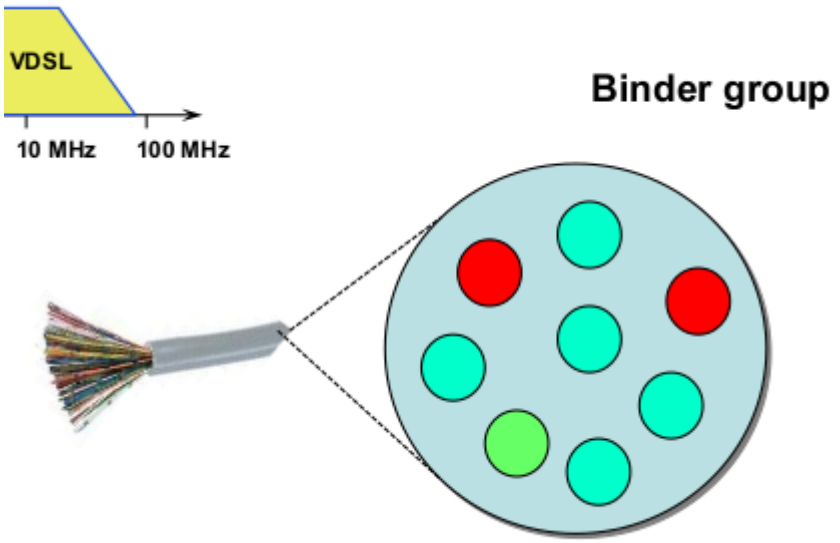
For doing modulation we do ***water filling***



If we image the channel as a pool that has a function that is the inverse of the attenuation (SNR^-1). The shape of the pool is the behavior of the channel in frequency. The lower this value is, the higher the SNR. As we know, we splitted the pool in different tones. The set of bits we want to transmit are the liquid that is put in the pool. The Water Filling algorithm puts more liquid where the pool is deeper. So if there is a part in which we can't transmit the pool

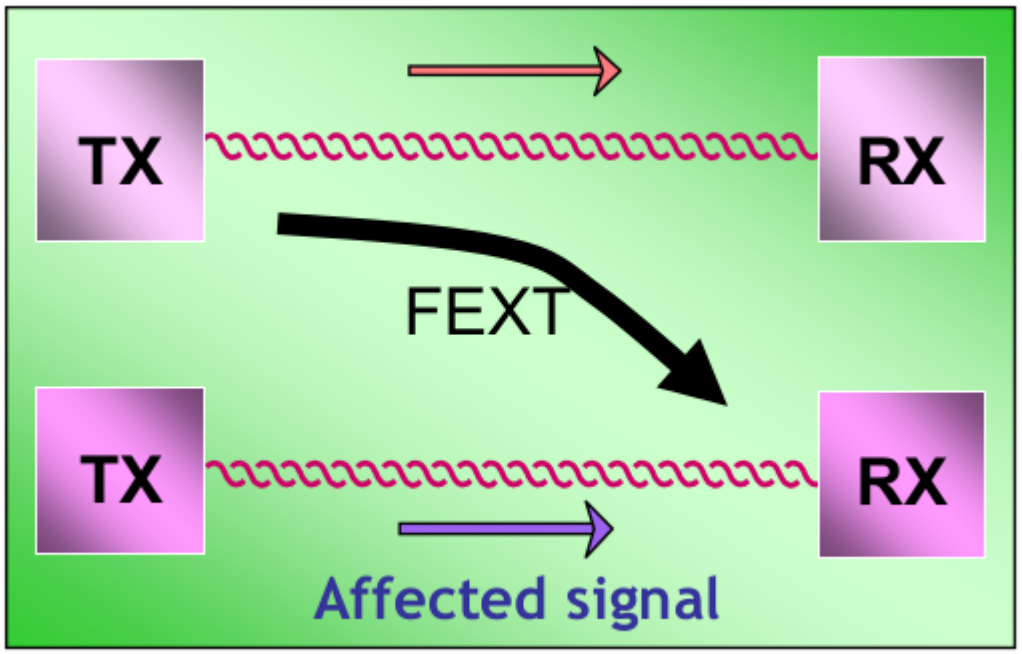
is very high in that point, as if there is a rock there, and less liquid will be there, so we will have a lower bitrate in that frequency.

Cross Talk: FEXT and NEXT



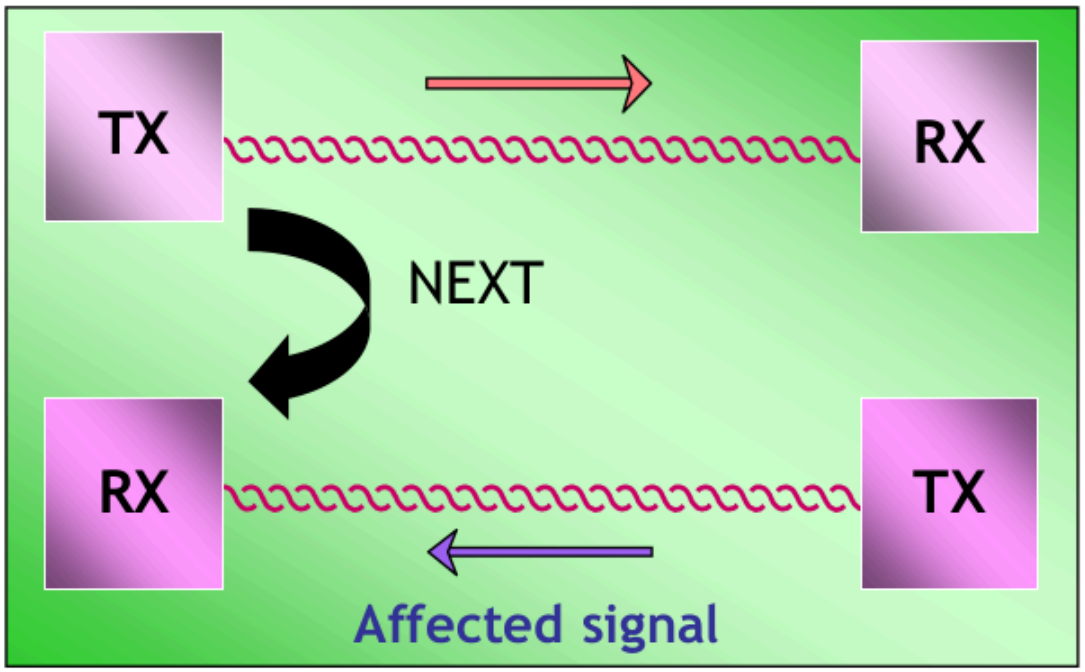
in the distribution network the private cables are aggregated all together in a binder group. The problem is that the higher the frequency the more probable an interference is from a cable to another, since they are close in space so the electromagnetic behavior makes some signal crossing from one cable to another. This was not present in the telephone, since the frequencies were not so high, but when we pushed to higher frequencies with DLS the problem arose. This interference is named cross-talk (XT) and is measured at the receiver.

Let's consider the case of **FEXT**:

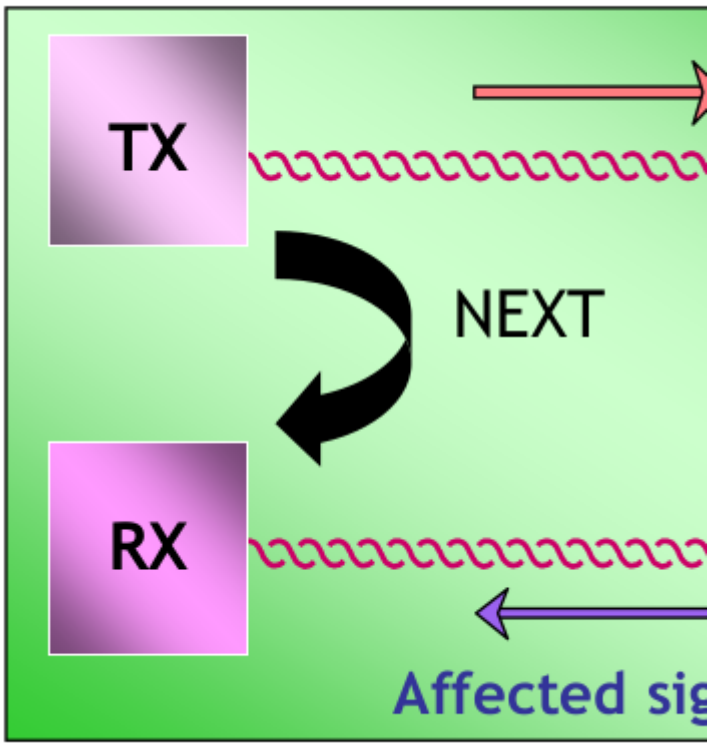


This is for downstream signals, indeed we have CO on the left and each RX on the right is a house. Is named “Far-End” since the interference happens very far from the receiver, and since a signal starting from the transmitter attenuates as a proportional function of the distance, the interference is not very critical.

The second kind of interference is **NEXT.** Remember that the interference is always measured at the receiver



But here, at the CO:



We can have interference, and this time is critical since the distance is short so there is little attenuation. If TX and RX are separated in frequency the NEXT does not exist, otherwise there is this interference. With Echo Cancellation we can recover and minimize the problem with NEXT, since TX and RX are in the same entity so we know what we transmit and receive and we can cancel one from the other, so do ECO Cancellation.

Another example, if we have RX and RX in CO and TX and TX in end-users, so a upstream signal, we have FEXT, but not NEXT

NEXT: POSSIBILE SIA NEL CO SIA NEGLI END USER -> NON È GRAVE NEL CO PERCHÉ POSSO FARE ECO CANCELLATION, MENTRE NEGLI END USER NO
FEXT: POSSIBILE SIA NEL CO SIA NEGLI END USER

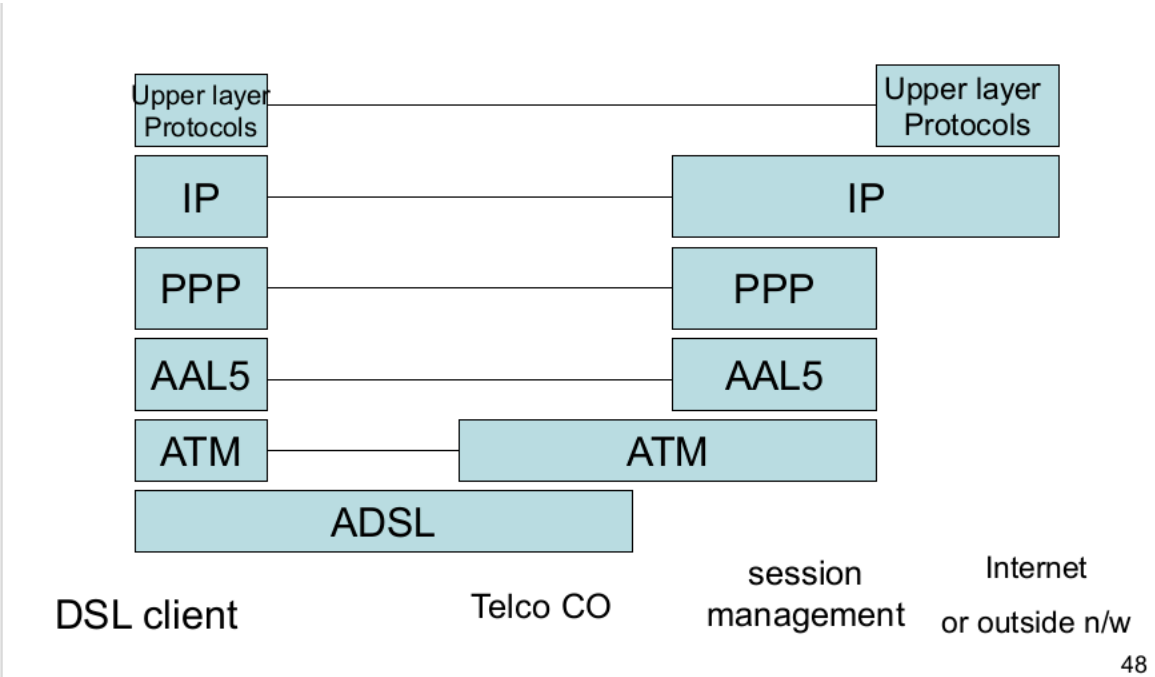
ADSL Protocol Architecture:

We can indicate that an ADLS architecture have for sure 2 interfaces:

1. from the end user to the CO: $I_{E\text{-}CO}$ (interface end user - central office). This is the external part of the net.
2. I_{mod} (interface from the modem to our final devices, like our smartphone) this can be wi-fi, ethernet, powerline

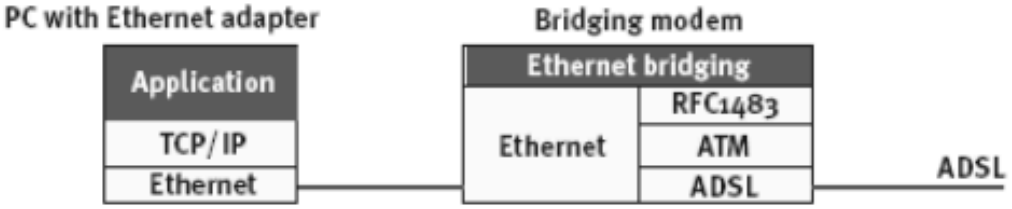
So our ADSL modem is a sort of router with 2 interfaces 1 toward the internal part of the house 1 toward the external part

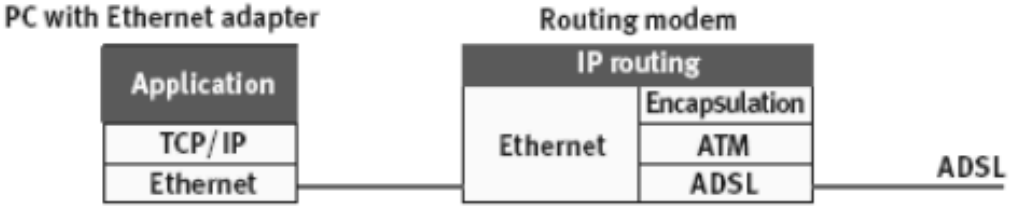
$I_{E\text{-}CO}$ has the following protocol architecture:

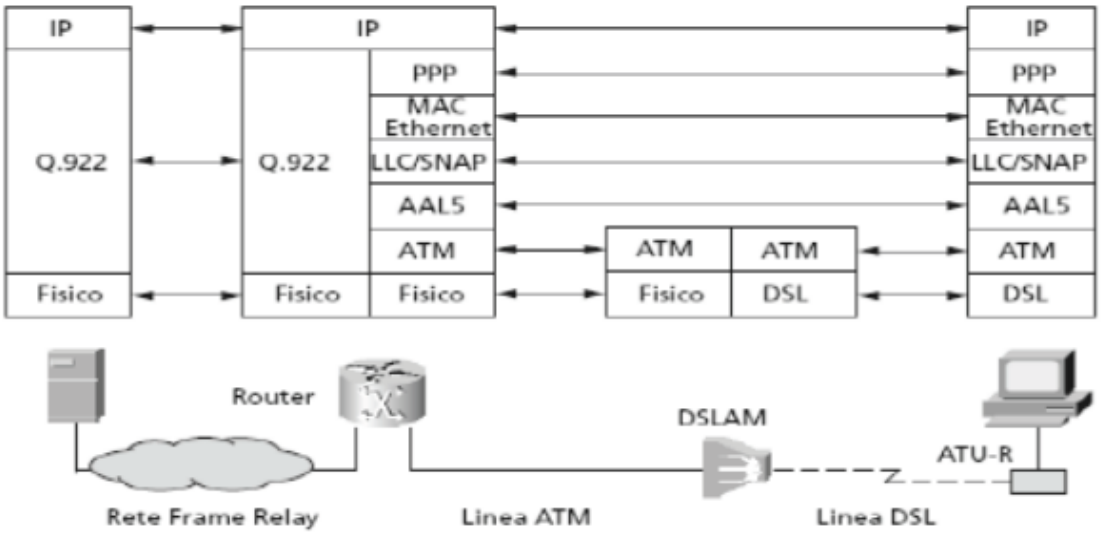


1. At the lower level (layer 1) we have ADSL. All operations relevant to the signal are done here, is a way to transport in a suitable way the communication in the cable.
2. Layer 2: ATM, support in a broadband way communication
3. Pseudo mac layer: PPP (Point to Point Protocol)
4. IP Protocol
5. Upper Layer protocol

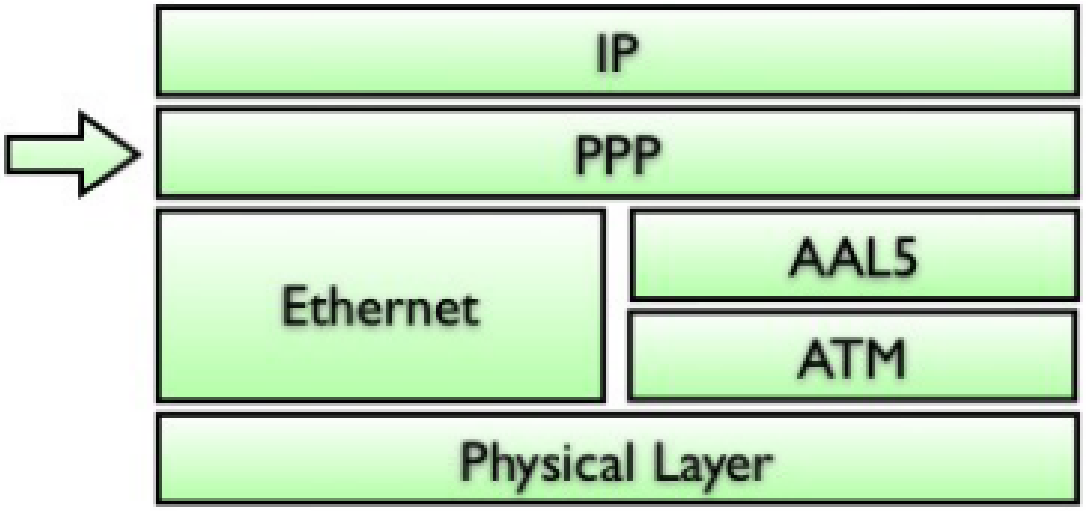
Whereas, on the user side we use TCP/IP:





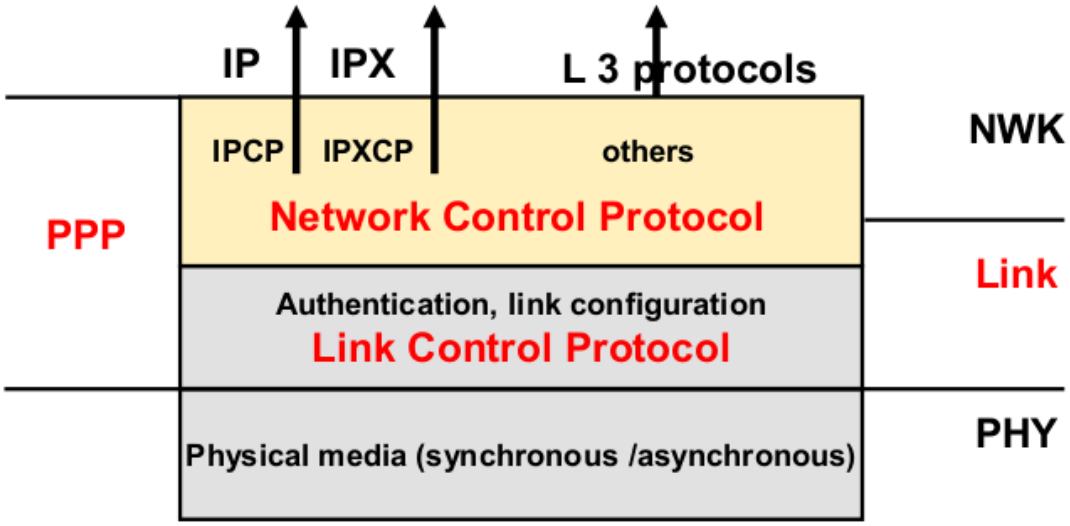


Point to Point Protocol (PPP):
Directly connect the user to the CO and provide authentication, authorization, automatic configuration of the interfaces and DHCP support. PPP can be encapsulated in AAL5/ATM or in Ethernet.



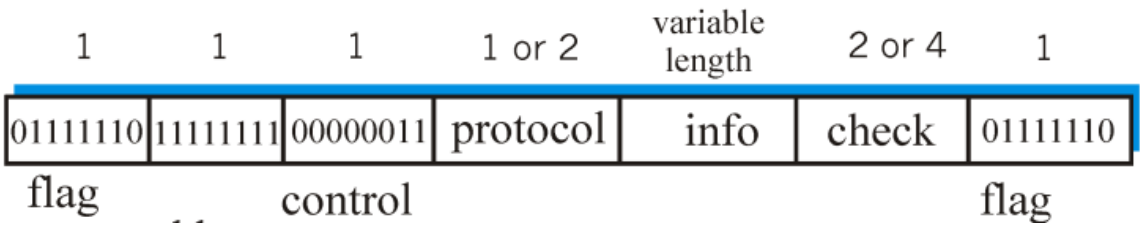
PPP has two components:

1. Network Control Protocol (NCP): configure network layers. Management of the public IP from the client to the net[11].
2. Link Control Protocol (LCP): establishes, controls, authenticates and terminates the link. So basically control the transmission of the system. It also does compression: the more we compress the better since this interface uses little bandwidth. So we need the info to perform the compression.



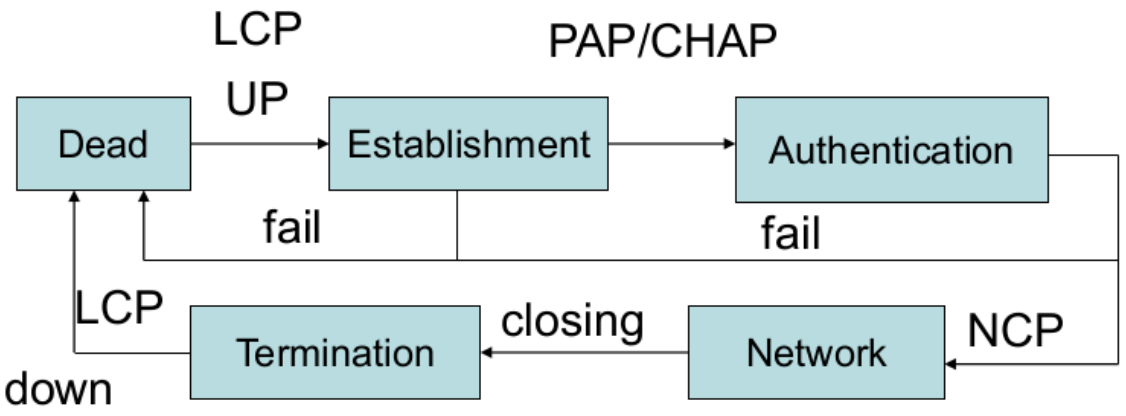
PPP frame:

[11] note that this is public because it is external to the home. In the home we use private addresses.



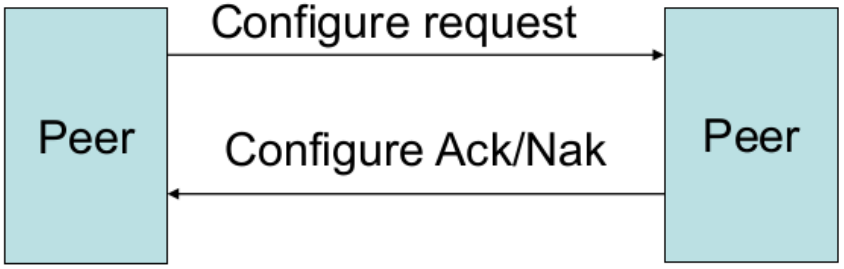
1. Flags: to identify the starting and ending of the frame.
2. Address: is not used, since is a p2p communication. Since it is not useful we set that to broadcast (all 1s) for convention.
3. Control
4. Protocol
5. Info: payload.
6. Check: used to verify the consistency, so if data are correct, since DSL is not so robust in terms of error.

Let's see the PPP operation:



LCP establishes the connection then we have authentication with PAP or CHAP. If the auth goes well, NCP establishes the network protocols. When we want to terminate, LCP tears down the connection.

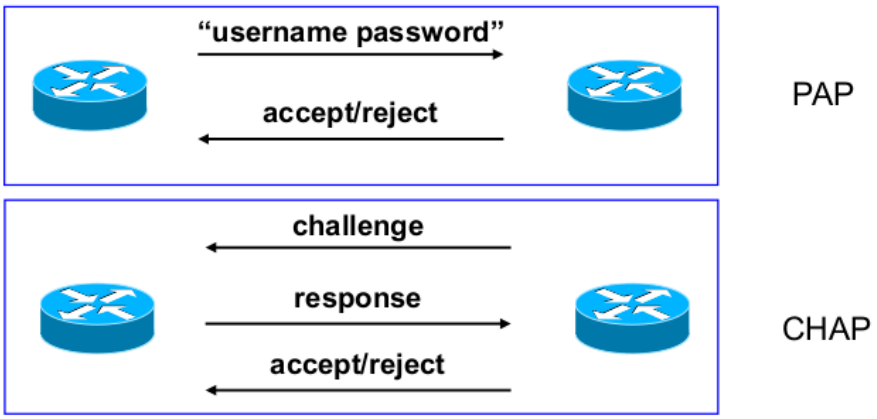
1. Link establishment



A Configure Request message is sent to request the establishment to the other peer. This message contains also the various options requested. If the peer responds

positively, it sends a Configure Ack message to accept the negotiation. Otherwise a Configure Nack message is sent together with an acceptable negotiation suggested.

2. Authentication:

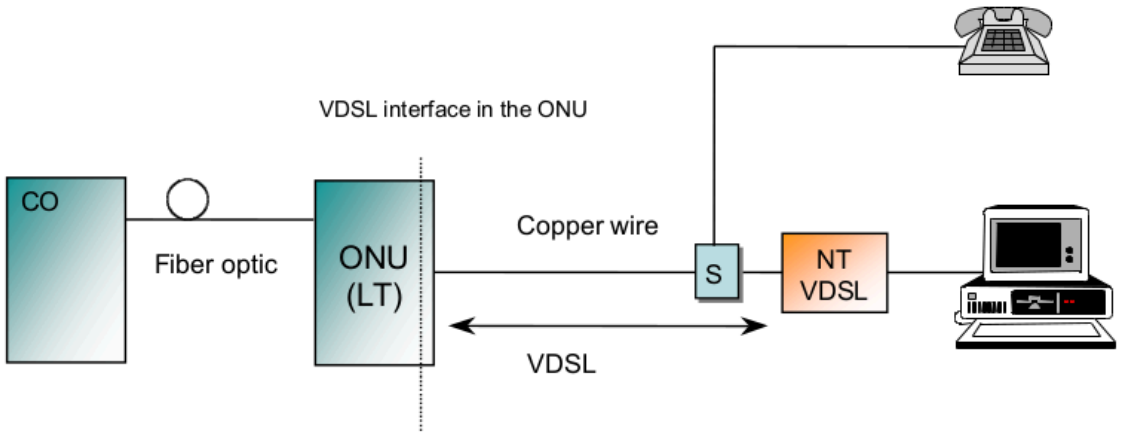


Two ways to authenticate:

- Password Authentication Protocol (PAP): using password for accessing the DSL. Not robust since the password is sent on clear on the line.
- Challenge Handshake Authentication Protocol (CHAP): the password is not exchanged between the two but they exchange a challenge, a random string of characters generated by the authenticator. The PW is already known by the two but never exchanged. The process is the following:
    - The authenticator sends the challenge to the client
    - The client uses a one-way hash function on a combination of the challenge and the PW, then sends the hash value back to the authenticator
    - The authenticator in the meantime also calculates the expected hash value with the challenge and its own copy of the PW. When it receives the client hash, it compares the two and if they match the authentication is declared successful. In case of succeed of the authentication we have the negotiation of parameters

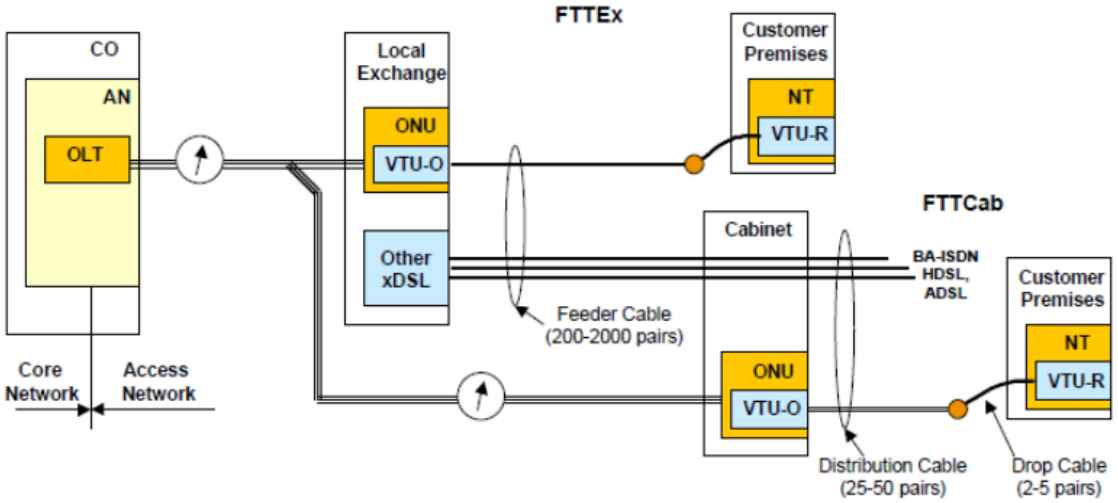
Very high bit-rate Subscriber Line (VDSL):
To improve DSL. The SNR decreases as a function of the distance, so VDSL wants to reduce the length of the copper, so to put the modem closer to the user. Previous to VDSL, as we saw, the modem was in the CO, with a length of the link of 1500 meters. This is a maximum length in case of DSL: at this distance it works at the nominal way. With VDSL the

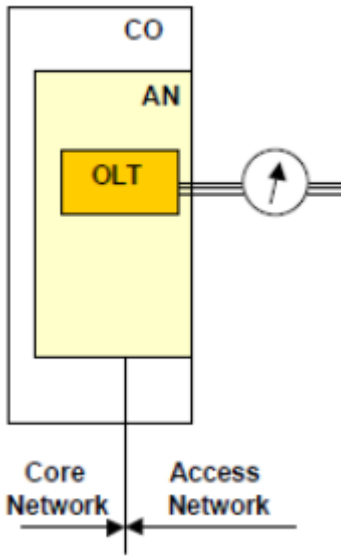
ONU goes further, and between the CO and the ONU we don't have copper but fiber.



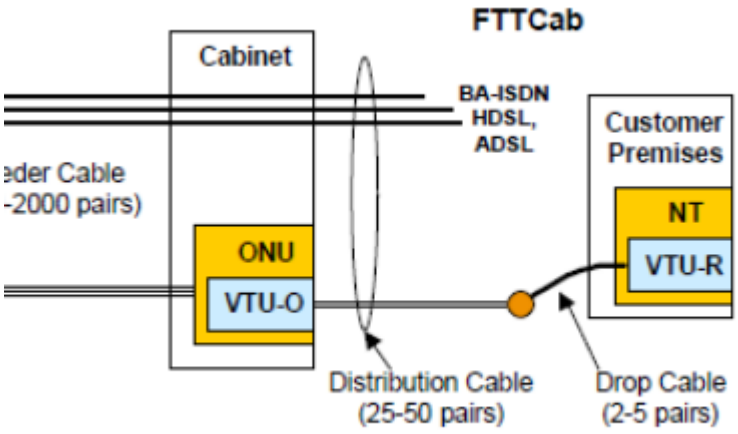
This was the first idea. This was done but has a problem: the deployment of fiber optic is expensive. We need pieces of our cities in which we have to put cables and ONU, and also we have to pay the workers and so on…

Typical Configuration of our cities, with a mix of FTTE, FTTCab…
We assume that the COs are what divide the access network from the core network.





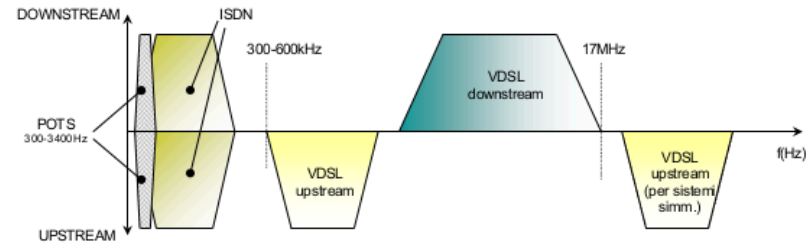
this above is the classical FTTE



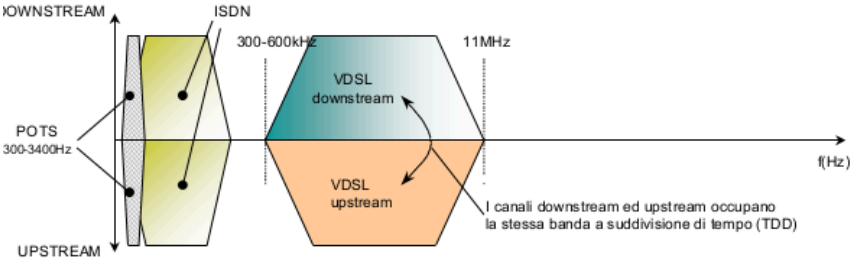
This above is when the termination is closer to the end user, with FTTCab. In this case they did the digging and all that stuff to put closer to us the termination.

This was the first solution for VDSL: to put the fiber termination closer to the end-user. Now let's talk about the second solution: use higher frequencies in the copper. As we know ADSL arrives at 1 Mhz, with VDSL we explore higher frequencies. We can do that in two ways:

1. FDD: so separated portion for upstream and downstream. Since they are bigger in areas they have bigger bit-rate
2. TDD: part of the time is upstream, part of the time is downstream. In this way we use the full bandwidth for both.



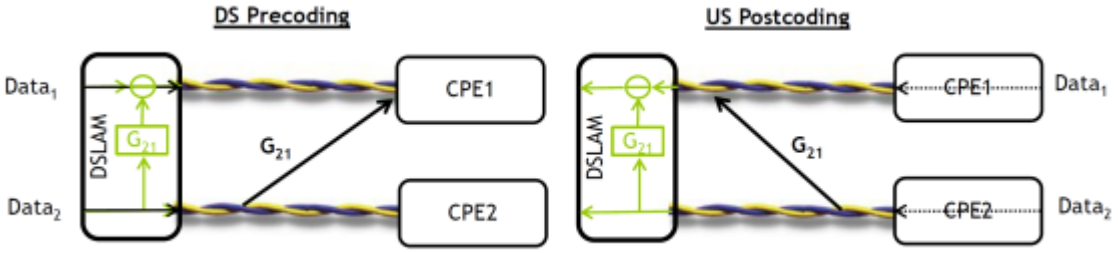
FDD solution



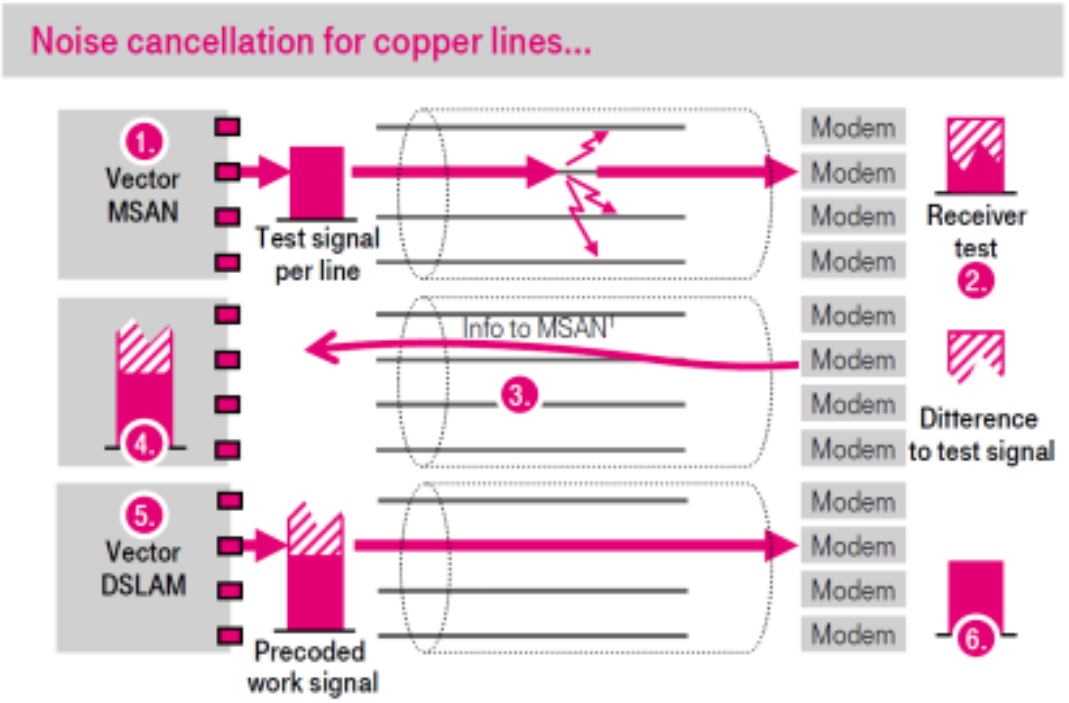
TDD solution

Vectoring:

Used in VDSL to solve the cross-talk problem. Vectoring is a smarter eco cancellation. To have a high bit rate, we want high SNR and high Bandwidth. The problem in cross-talk is in SNR, since we can have interference. This time we can't neglect FEXT, since with VDSL the ONU is closer to the user so the cable is shorter and so the interference is not attenuated enough while transmitting the signal.



Let's analyze a specific user in the binder group, that has its own signal and the interference from the other user's cables. My goal is to obtain the amount of interference and how it affects my channel, but how can I do that? I can do probing: the transmitter sends a well known signal, the receiver, that knows this signal, can compare the received one with the known one and say "no I received the signal completely modified" and so I can measure the interference and the real shape of the channel. Then the receiver sends back the information about the distortion to the transmitter now the transmitter knows how the channel is done since it has discovered how the channel distorts the signal. Now the transmitter can retry the transmission adding to the signal an *anti-signal*, that compensate for the distortion of the channel. To provide vectoring, we have to put a computer that does the job in the cabinet.



At the very beginning this vectoring was not implemented in all binder groups, so we had a partial vectoring only for some lines.

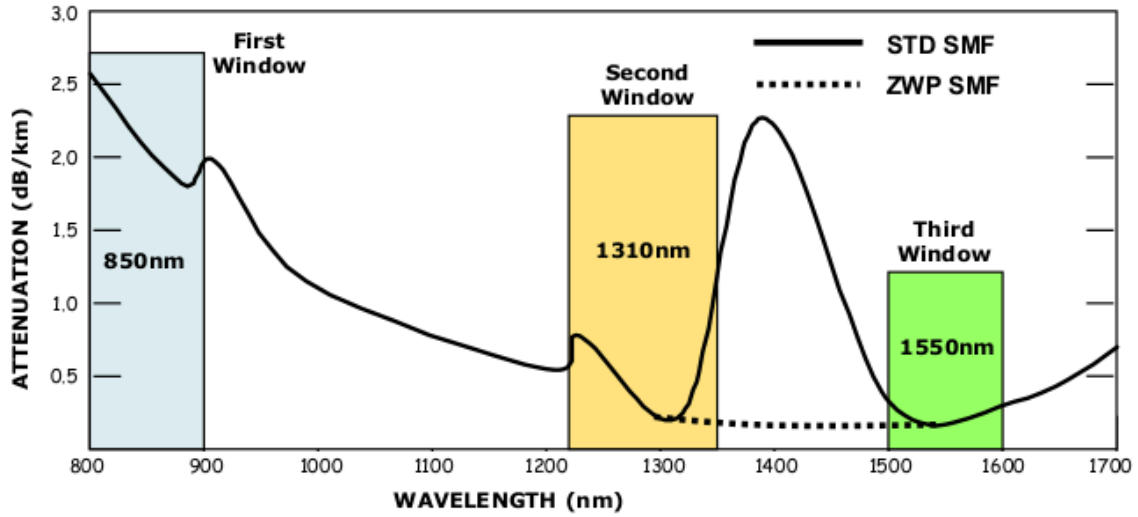
Passive Optical Network

20/11, 27/11

The transmission in the fiber happens thanks to a very punctual laser, on the other side there is a photodiode that captures the light and transforms that into an electrical signal.

Optical Fiber Attenuation

This plot shows the attenuation behavior for fiber cables in the different ranges of wavelengths:

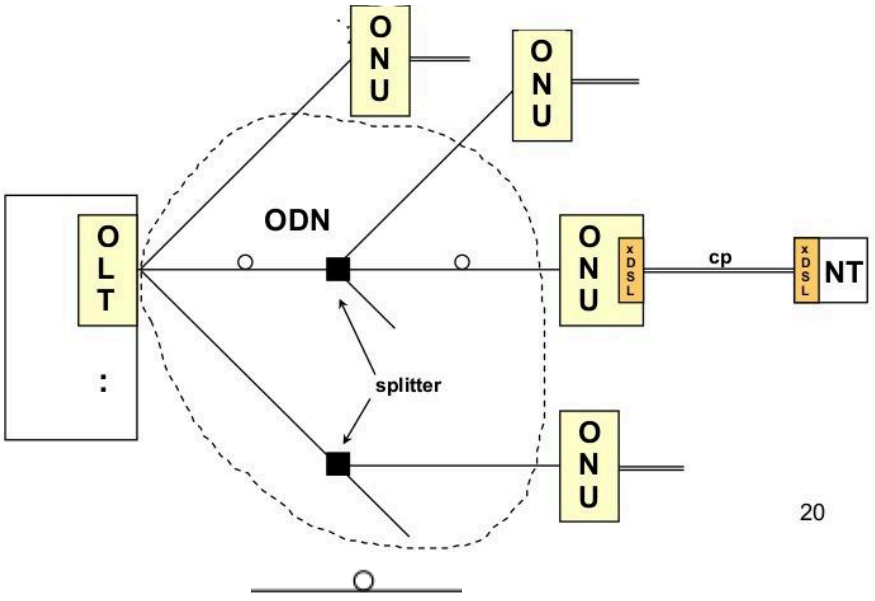


There are some windows of the wavelength spectrum that are preferred to pick up the wavelength for transmission since they behave well in terms of attenuation and shape of the channel.

Fiber Optic Access

High quality in bit rate, reliability, latency, and it's quite secure. The FTTx Architectures have the following elements:

1. OLT manages the optical transmission
2. Optical Network Unit (ONU) is the one that is close to the user.
3. Optical Network Termination (ONT)
4. Optical Distribution Network (ODN)

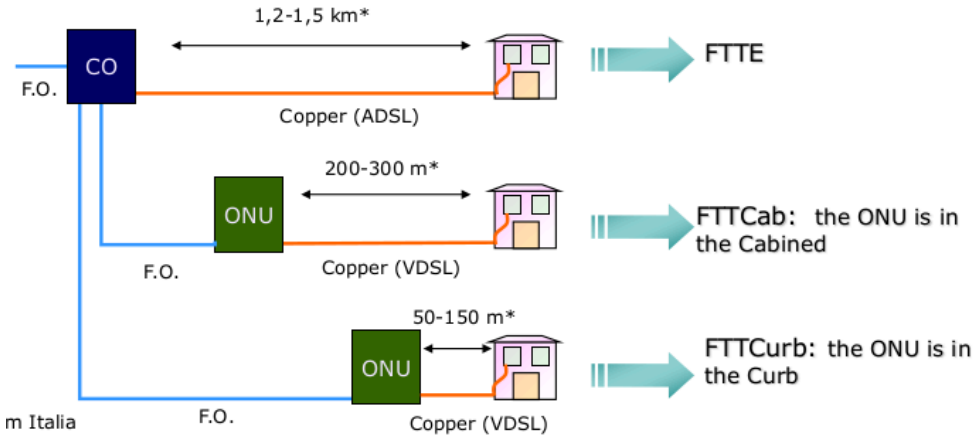


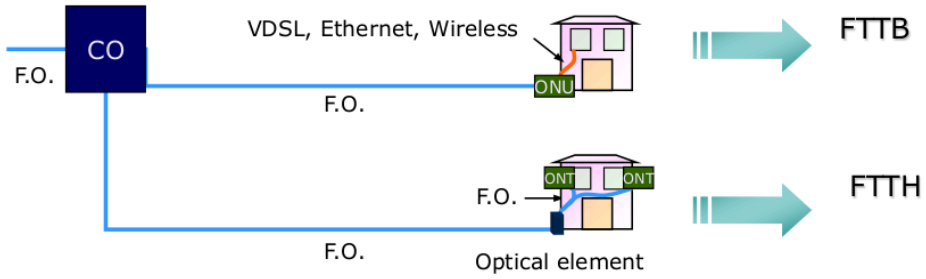
Note that this symbol: indicates a fiber cable. From the FTTx family, we can consider:

1. FTTCurb/FTTCab: the fiber arrives in the cabinets, that are between the buildings and the CO.
2. FTTNode (or Neighborhood)
3. FTTPremises. It's a generic terms for FTTH/FTTB:
    a. FTTBuilding (or Business)

b. FTTHome

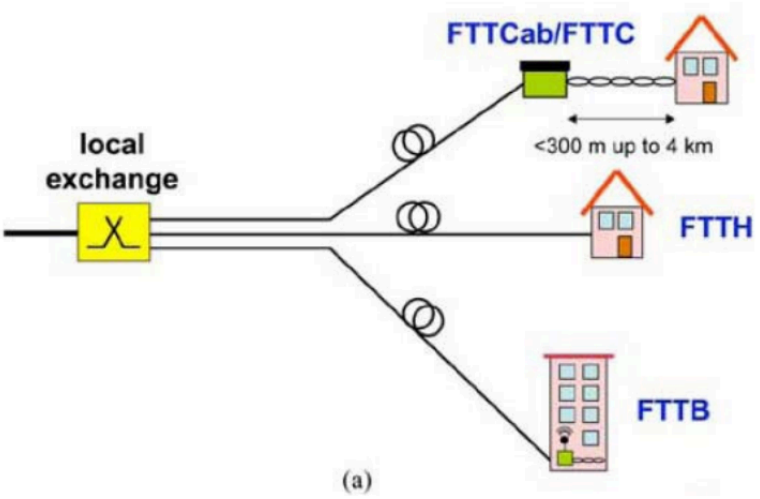
4. FTTExchange: the fiber arrives at the CO.

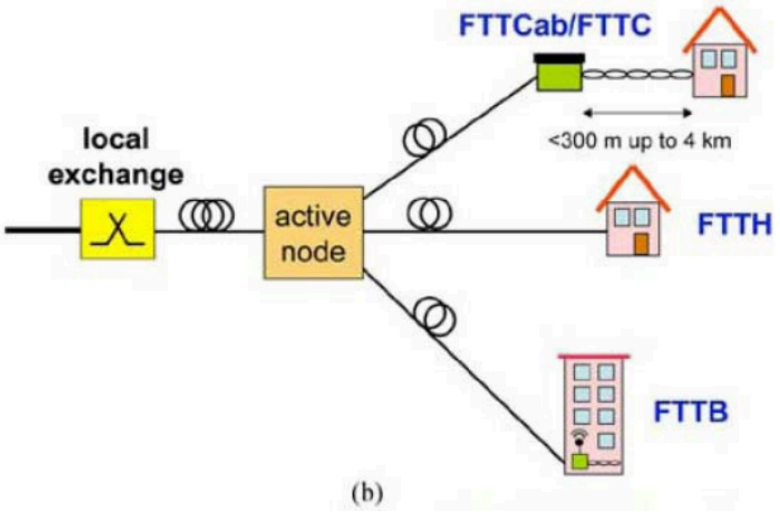




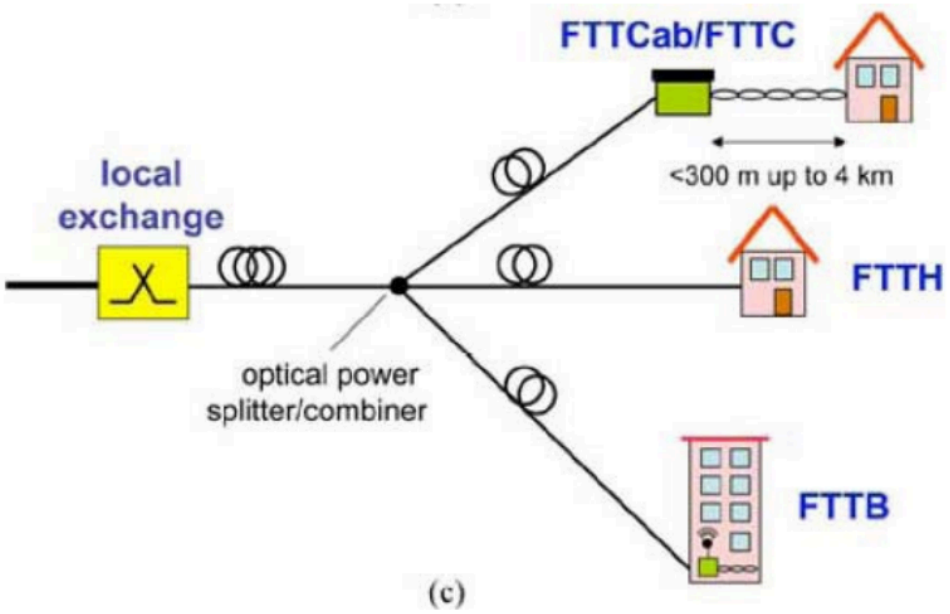
FTTx family is divided into:

1. AON: Active Optical Network (or Point-to-Point), in which there is an active switch that is switching the data. We can have a separation of cables from the local exchange or after an active node, closer to the end-user.

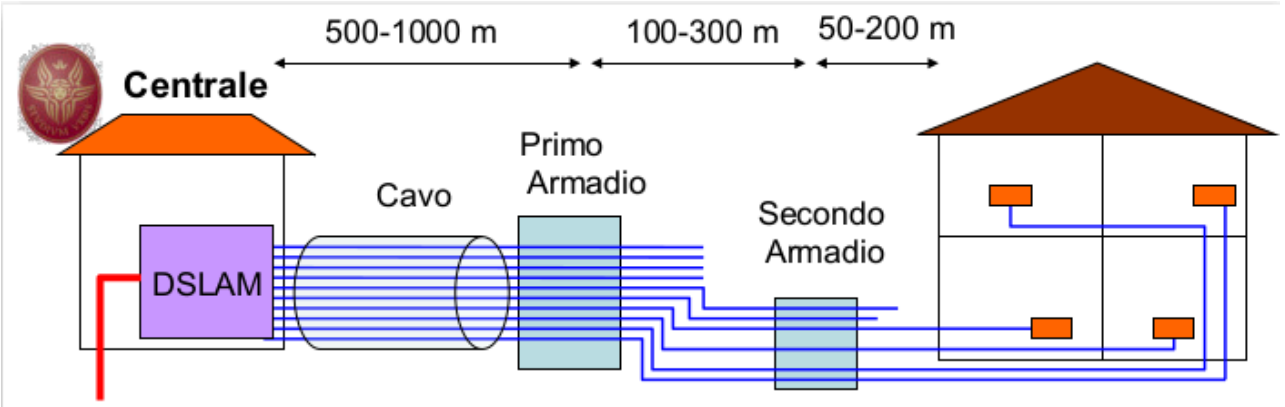




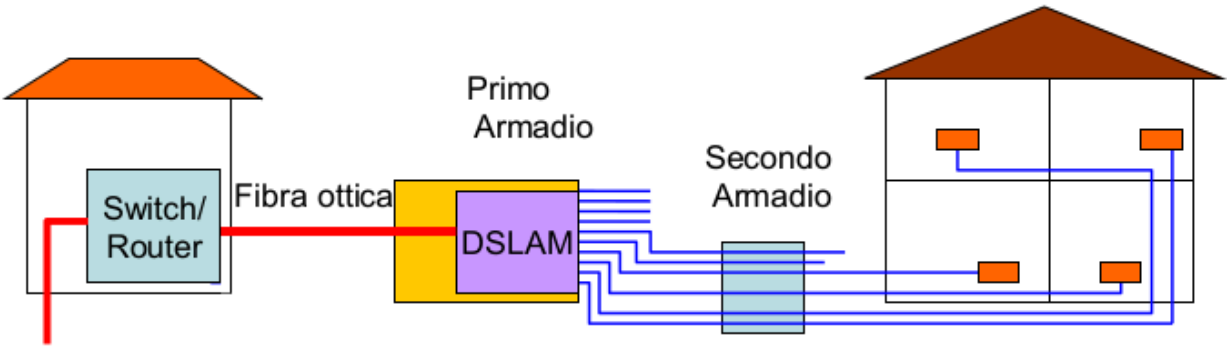
2. PON: Passive Optical Network: there is no switching, just the signal is splitted in all directions, there is no routing, everyone receives the signal of the other. Active ones are costful, while passive ones cost less.



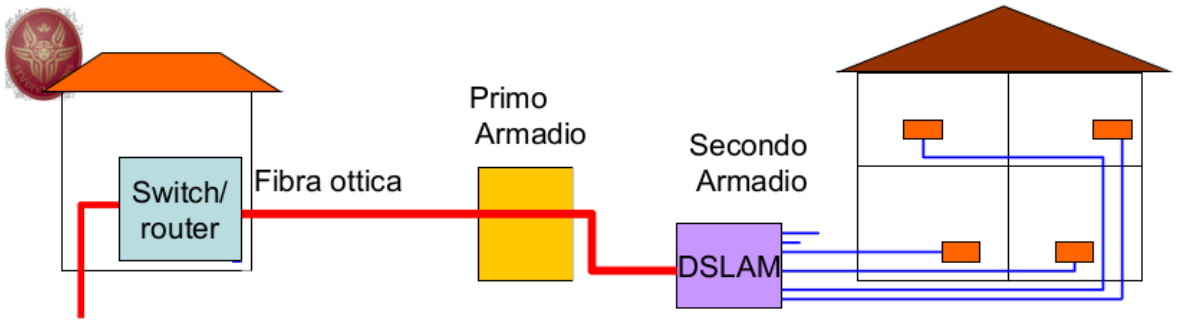
If we want to put the fiber closer to the user, we have to put the DSLAM closer too, until the FTTH in which the DSLAM is not present anymore



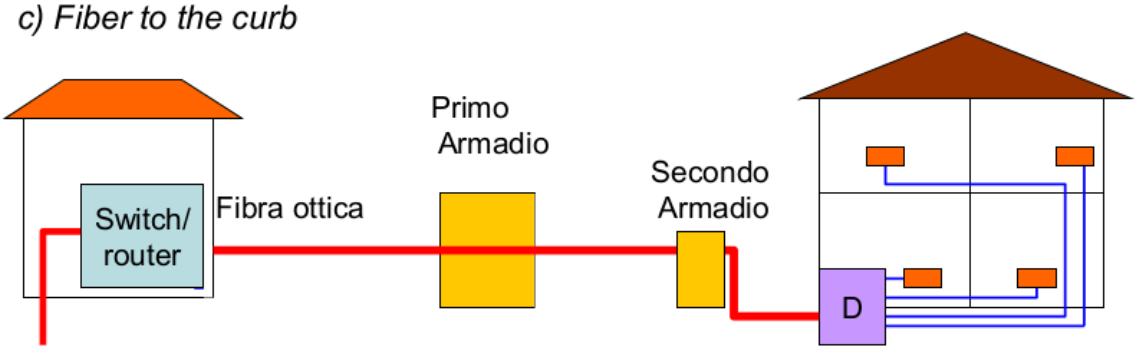
a) Best current architecture



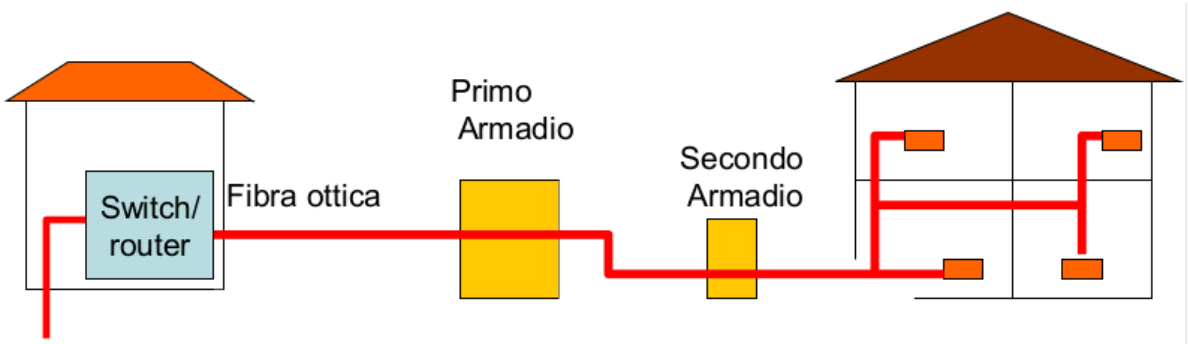
b) Fiber to the cabinet



c) Fiber to the curb

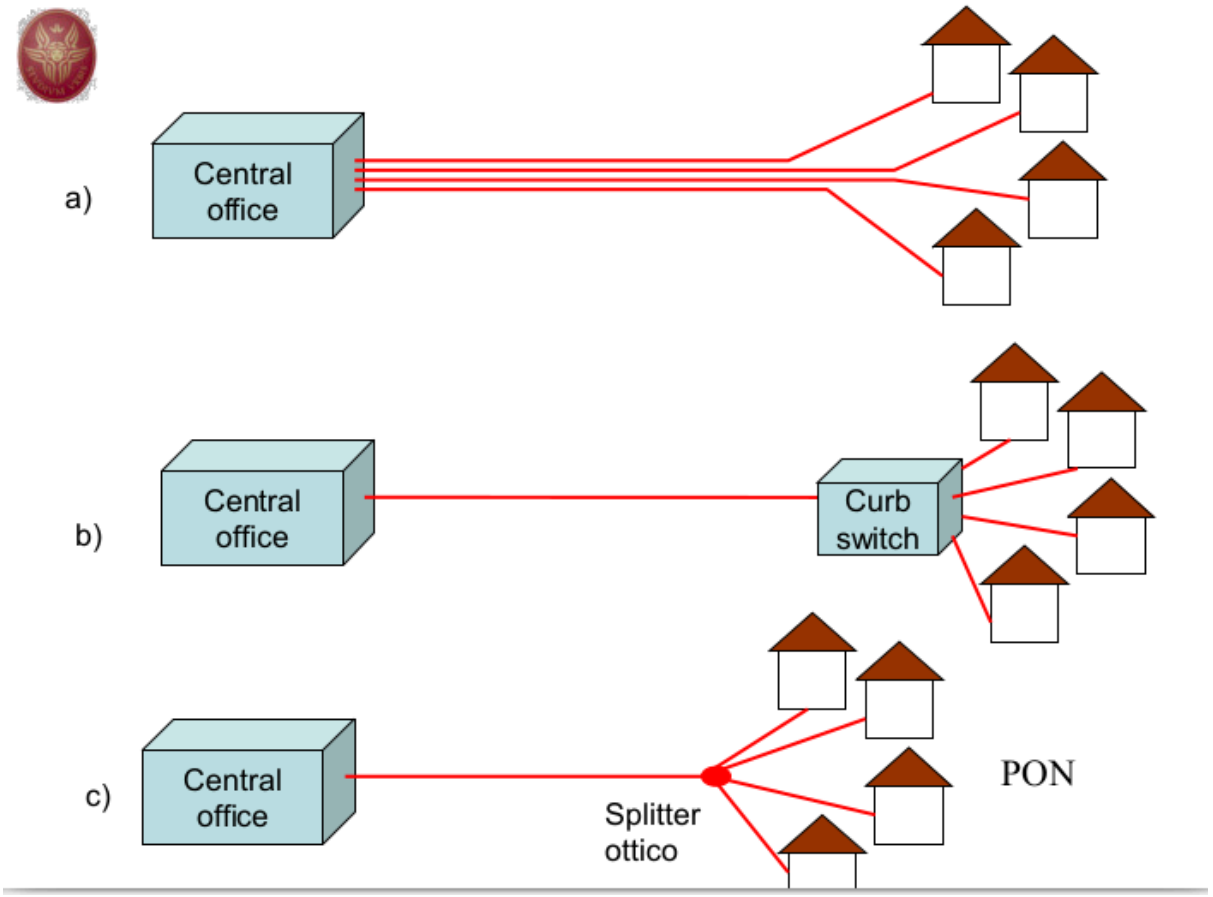


d) Fiber to the building



e) Fiber to the home

How to distribute the Fiber? We have three ways. Let's consider for instance the case above in which we are arriving with the fiber to the end-user, so it's FTTH, but we use this approach for distributing the fiber also with the other member of FTTx.

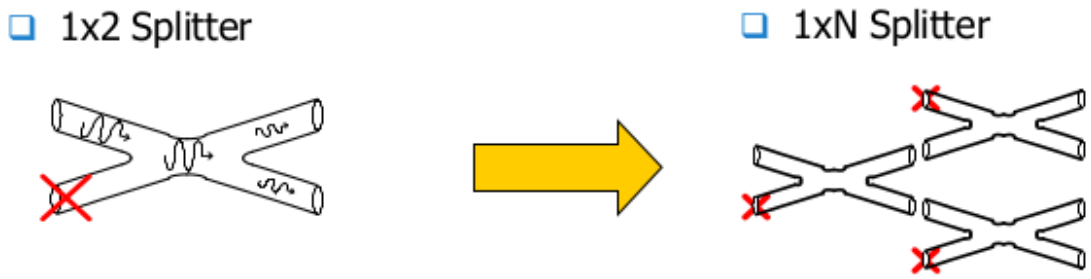


OLT, in the CO, is where the fiber starts. The best case is FTTH, in which the ONU is at the end user.

1. First possibility, each home has a ONU and each ONU is interconnected to the CO or central element by having 1 fiber, so n user n fiber. This is a sort of star topology. It is costful since we have to dig n fiber cable.
2. Second: fiber until an element that is a sort of switch in the cabinet/curb, then the communication is splitted for the different users, still in fiber even after the switch.

Advantages: only one cable until the curb, so 1 cable 1 digging: less costful. The disadvantage: we have a net element that occupies spaces and also, since the switch performs intelligent operation, we have elaboration typically in the electrical domain. To elaborate we need power, so electricity to spend.

3. Third: passive optical network. I have a fiber until an element, the splitter. This is a passive splitter -> we don't need electricity to split the signal or space to put the switch so it costs less. This is not real splitting, is a forwarding of the full signal in all the directions. The splitter is a piece of glass



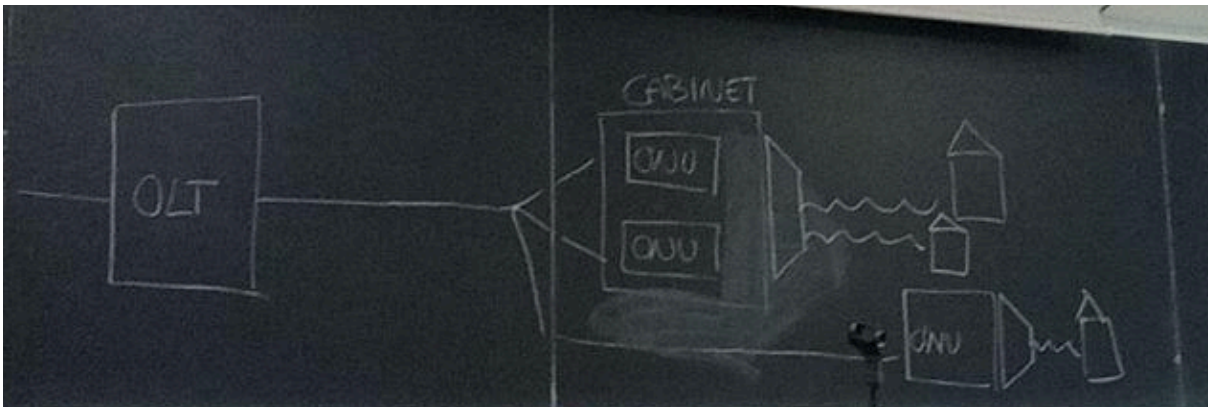
The simplest is 1 to 2, 1 ingress 2 egress. Obtained by fusings 2 fibers. 1 passage is closed. So it's not actually a switching, just a propagation. To do 1xn you just do that in cascade. In the image we have 1x4. Drawback: the pathloss (attenuation) during the splitting, so I lose some power, but it is few. As we said, this was in case we arrived at the end user (FTTH). If we arrive at a cabinet we can use the same architecture with passive splitters. The cabinet terminates the fiber optic and we start DSL with copper wire.

If we want the CO more distant to the user we can't lengthen the distance in copper more than 1,5 km, or we will lose bit-rate. So the DSLAM should be near the user to have better performance, and what we can do to lengthen the distance is to use fiber.

PON Network:
Let's analyze in depth a PON network. We have an OLT that is interconnected with the core network. If xDSL exists, is after the ONU, closer to the end user, as shown in the picture

below:

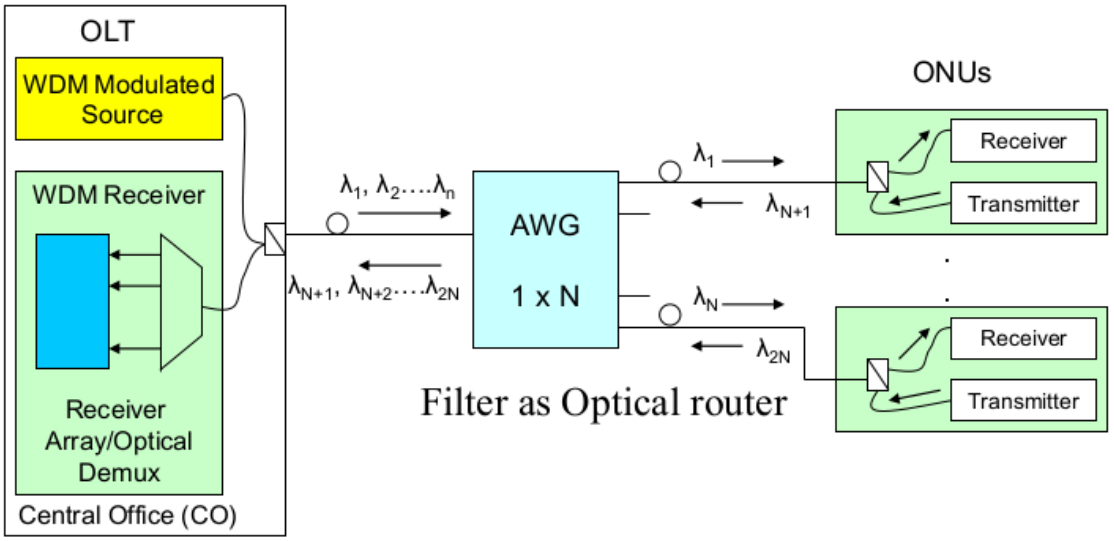
Let's see this example



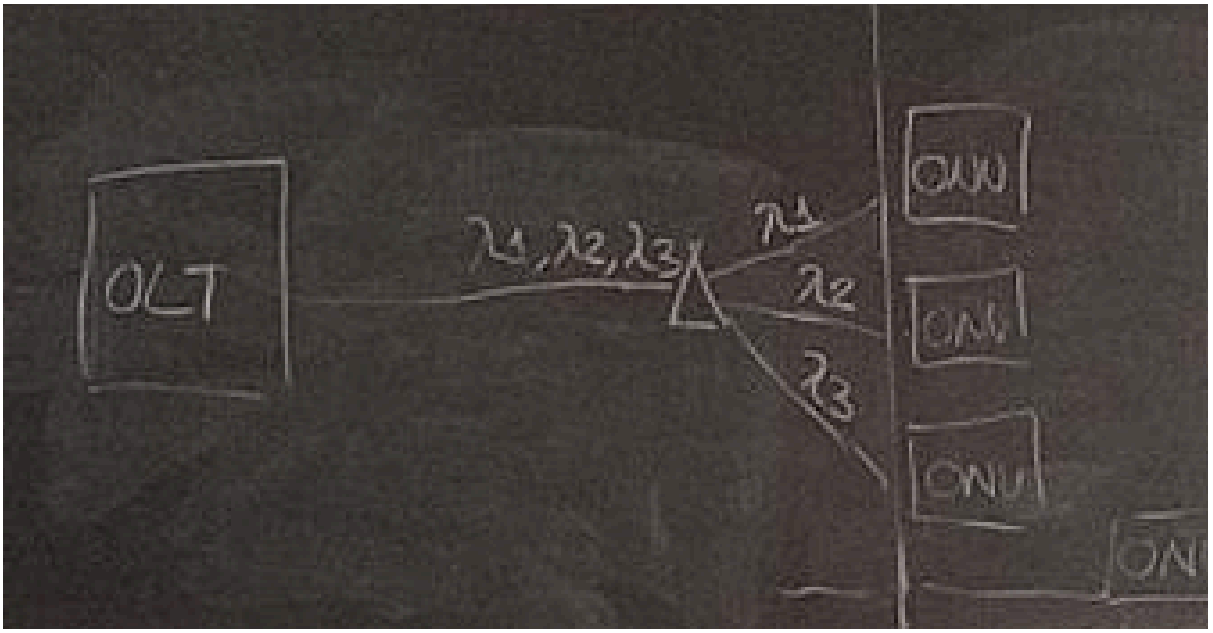
In this case above we have the cabinet for some users and the curb for other users. The part in which we have different branches of different fiber cables is the splitter (between the OLT and the cabinet/curb). The part from the OLT to the cabinet and curb can be very very long since it's fiber and not copper, so not 1,5 km as limit.

In the fiber cable, we can use wavelength of lambda = 1.5 nanometers from the OLT to the splitter. The same wavelength is used after the splitter until the cabinet/curb, since the splitter does not operate on the w.l. level. If the splitter was a switch it may happen that the w.l. is modified. We can differentiate the directions using 1 w.l. for upstream and 1 w.l for downstream. This is possible since we are doing Wavelength Division Multiple Access (***WDMA***). We could also do ***TDM*** and so just 1 w.l. part of the time is used in the downstream part in the upstream.

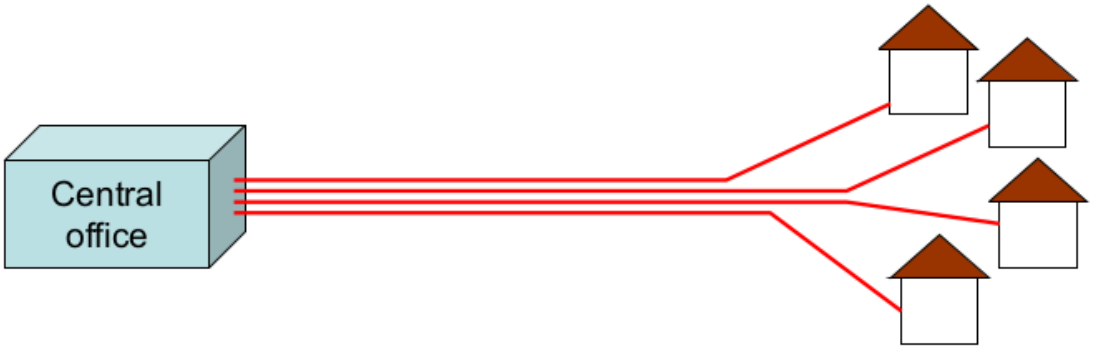
Example of **WDMA-PON**:



Today a fiber can transmit more wavelengths, thanks to lasers that are able to emit different colors, they have to be punctual to transmit, let's say red, blue and green precisely. Then at destination we have to pick up a single color from the medium. In the middle we have an optical router, a filter **AWG**, connected to the OLT with a single fiber cable. AWG is a passive element that has better capabilities than the passive one before and that splits the color as a prism does. Note that in the [OLT, AWG] cable we have n wavelengths if we have n ONUs that receives the signals.



AWG works in both directions (upstream and downstream). This is physical, if we want to represent that logically it’s like a star. We don’t have the problem of collisions anymore. So, thanks to AWG and this architecture, we came back to:



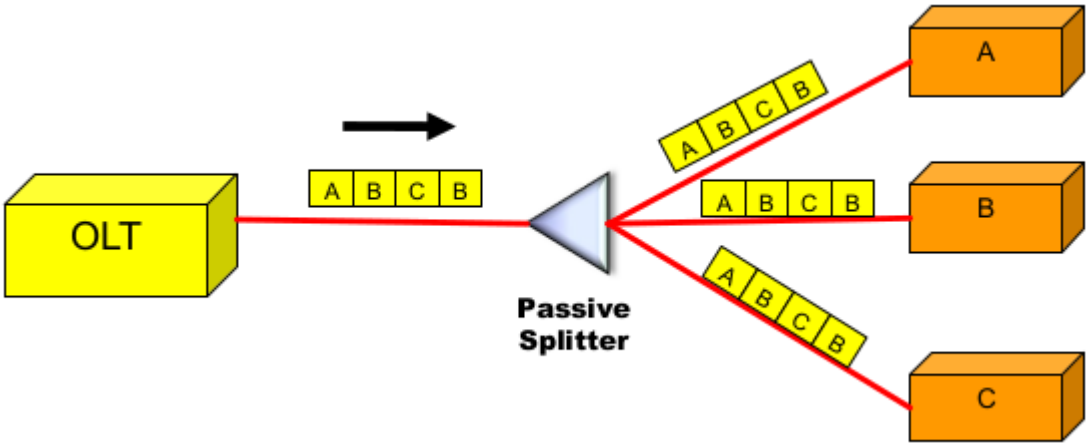
In which each user has its own media, because it has its own wavelength. We can say that the fiber cables from the AWG to the houses are *physically and logically separated*.

This architecture has however disadvantages:

1. if the shared cable is broken no one receives anything
2. if VODAFONE has some lambdas and TIM others and the cable fails, who is responsible for the cable?
3. the prism can't distinguish between an infinite number of w.l., so we have a limited number of w.l.. I have to improve the prism to divide better and have more w.l.

Downstream and Upstream Scheduling in a PON Network:

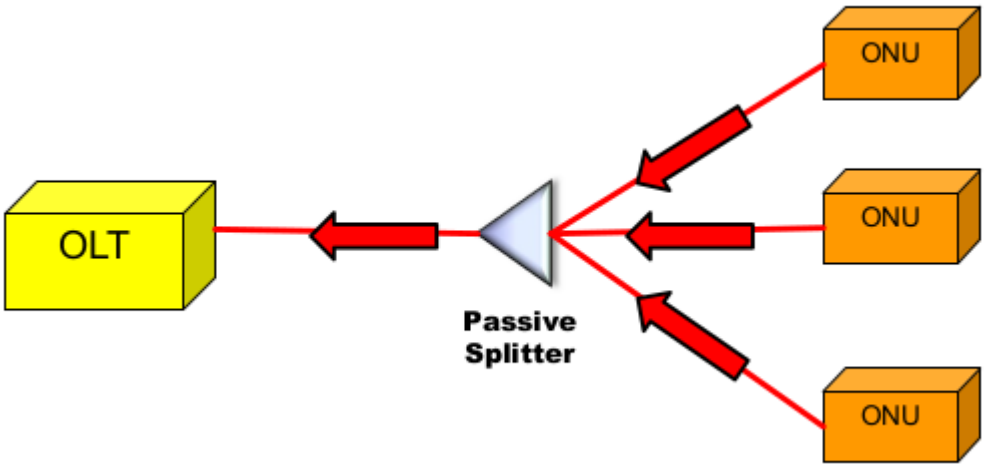
Let's see the downstream first. The signal arrives to everyone, so the splitter just lets everyone receive the same signal. So everyone can observe the signal of the other, unlike DLS, in which each user has a cable, so the communication was physically separated. How can I separate the info for ONU1 from the one for ONU2, ONU3 etc.? The solution is to put on top of the signal a layer 2 protocol able to distinguish the information and its destination. We can divide the channel in pieces, for instance in time (TDM), providing time slots.



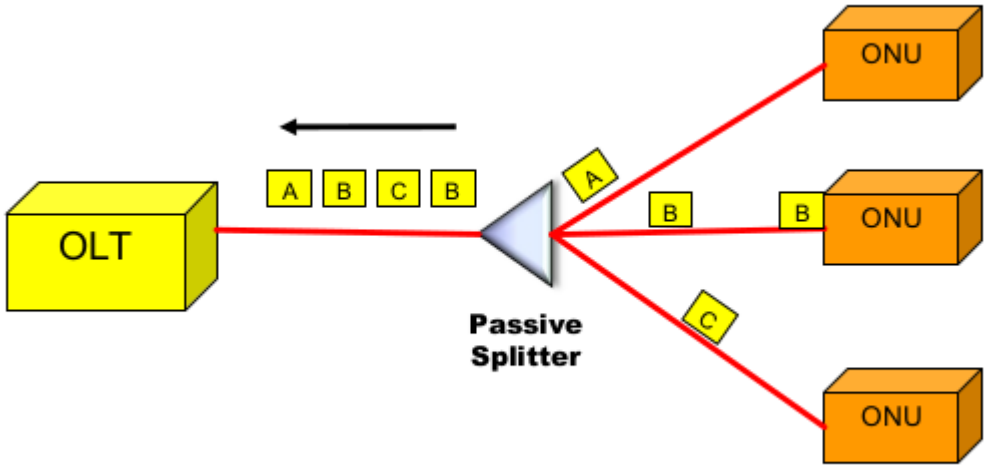
It's up to ONU_A to read info A and discard info B and C. The drawback of this approach is that we have a waste of resource (A receives also B and C info), moreover, we have more complexity since we need a protocol that decides this schedule in time: for instance, at a

given time another user arrives and becomes active, we have to put the space in the time slotting to accommodate its information. There are some similarities with ethernet: we have a shared medium and multiple users that transmit and receive all the information. The communication happens in a synchronized manner.

Nevertheless, this was the downstream, the easier part, let's see now the upstream. We can actually have the same identical configuration but we have to coordinate the communication in order to avoid collisions.



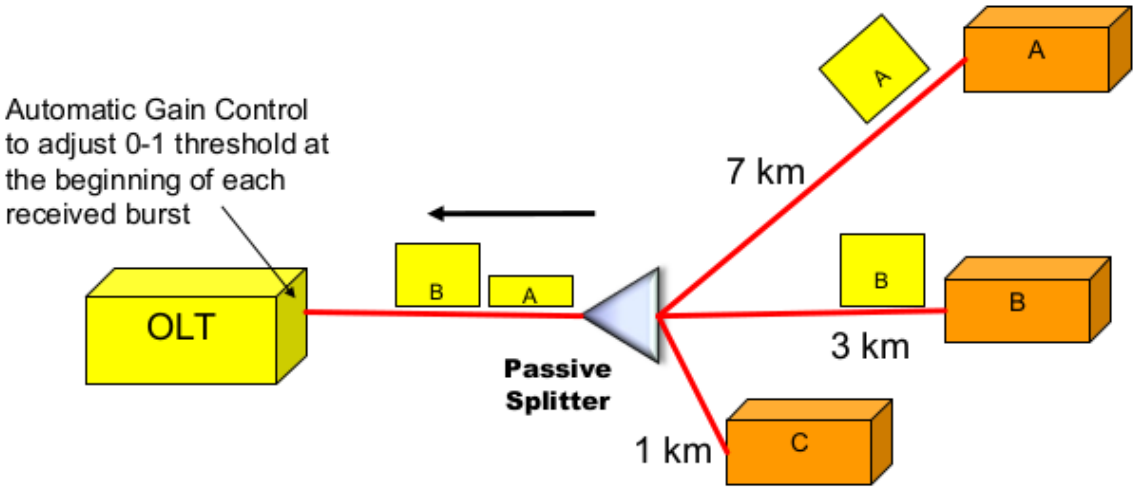
The ONU doesn't see the structure, they just see their part, it's not shared physically anymore. We need to share it logically even if physically it is not shared, otherwise there is the risk of collisions.



Since it's quite problematic to have perfect synchronization, we need to put some space, a sort of gap in time in order to avoid collisions. This is a drawback since we are not using resources.

Note that *we can't sense the channel as in Ethernet with CSMA*[12], since an ONU that senses its channel, since it is not shared (it's only logically shared, not physically), will always find it free.

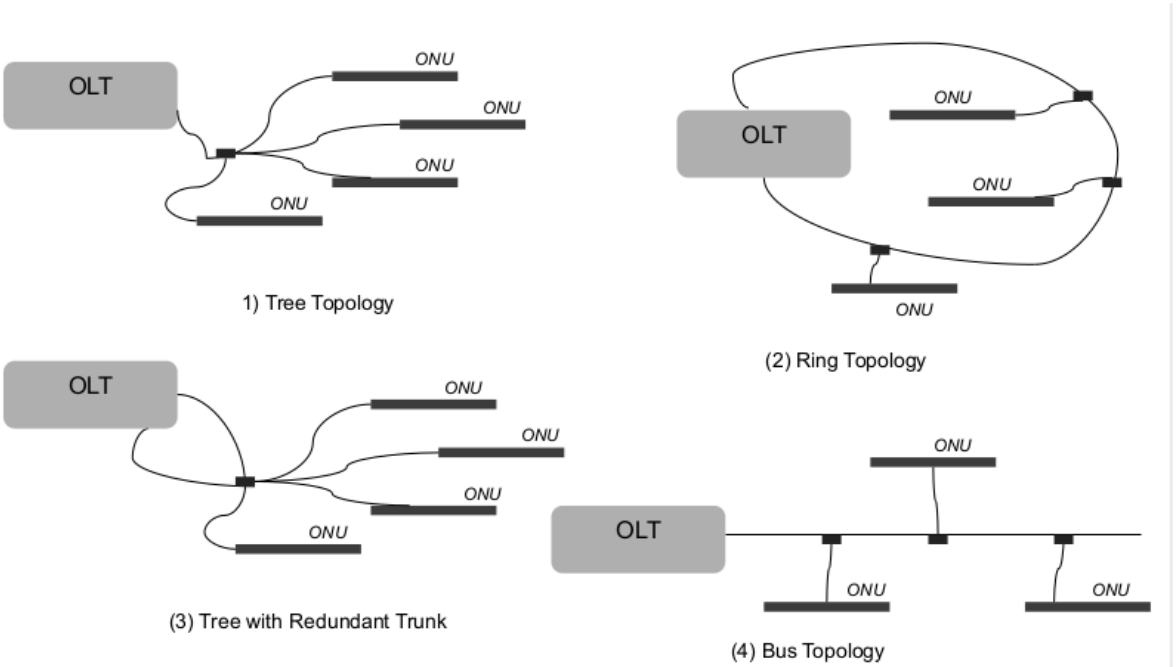
An important information we have to have is the length of the path [ONU,SPLITTER], since it's relevant for the timing to avoid collisions: assume we have A and B fully synchronized



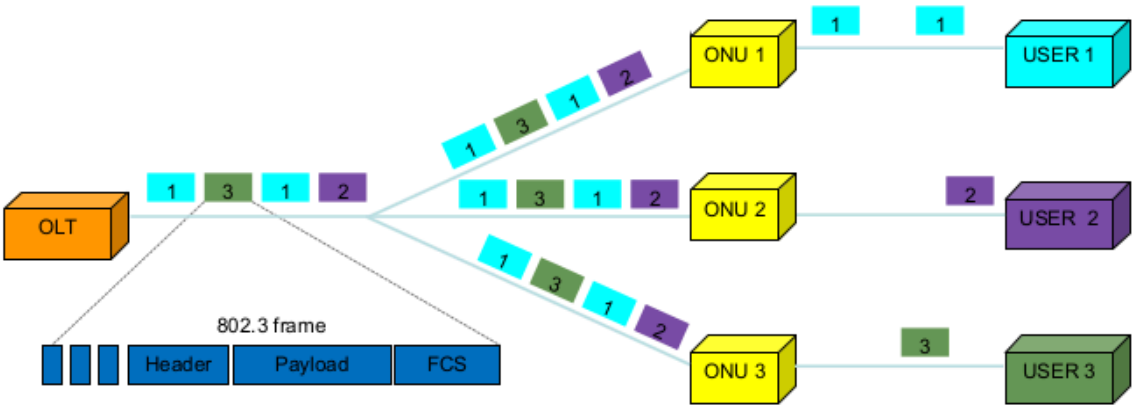
the length of the B branch is shorter than the branch A, the power of the info is represented by the height of the yellow square. If A and B messages have arrived at the OLT with different power (different height) the OLT sees oscillation of the power in the order of microseconds. If you have very high oscillation in a short time you can't recover signals. To solve this we have to send messages that arrive to the OLT with the same power. For doing so we use probing: give to the ONUs the information about how much power they have to transmit the message with in order to reach the OLT at the same identical power, so there will not be oscillation reading the packets in the OLT. In the picture above, A has to communicate with a higher power because it is more distant than B.

Ethernet Passive Optical Network (EPON):
EPON possible topologies:

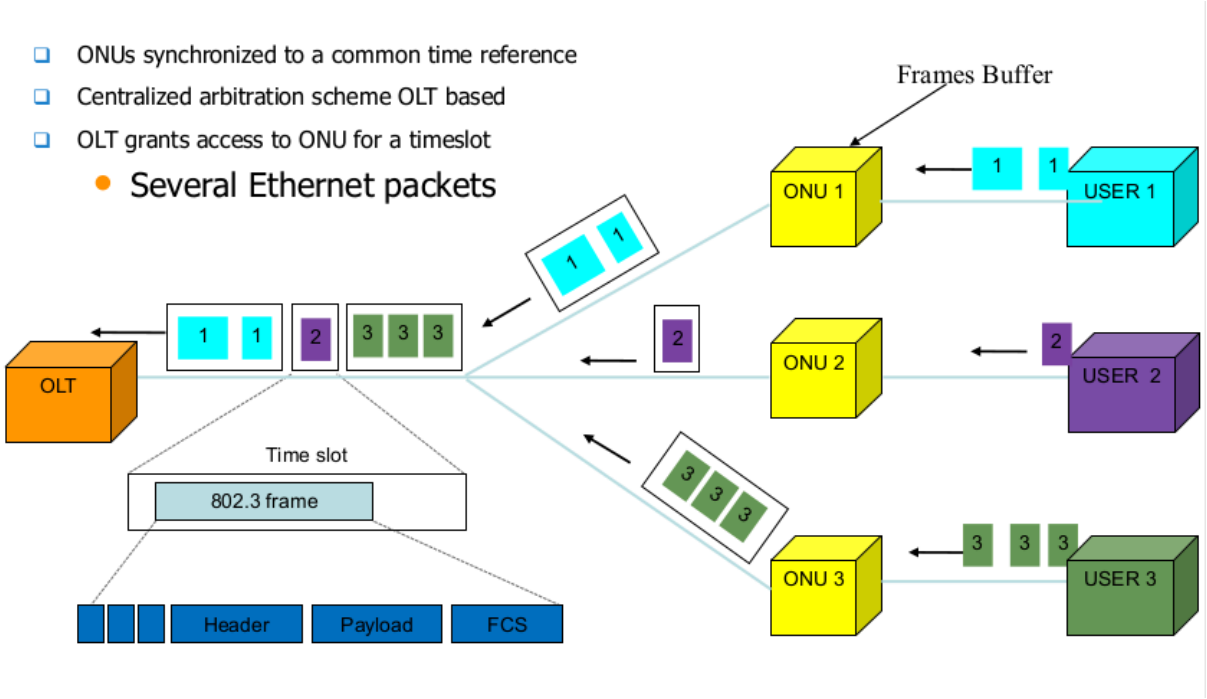
[12] Problem of multiple access on single media. CSMA solves this in the following way: before transmitting your info, you have to sense the medium to understand if it is occupied.



Let's see in depth the Tree Topology:



The stream of data is formed in a unique flow and in each piece of data I put the address of the intended receiver, as in ethernet. In this case we don't need time slots in the downstream, on the contrary in the upstream we should synchronize. The OLT grants access to ONU for a timeslot, so the ONUs are synchronized to a common time reference.



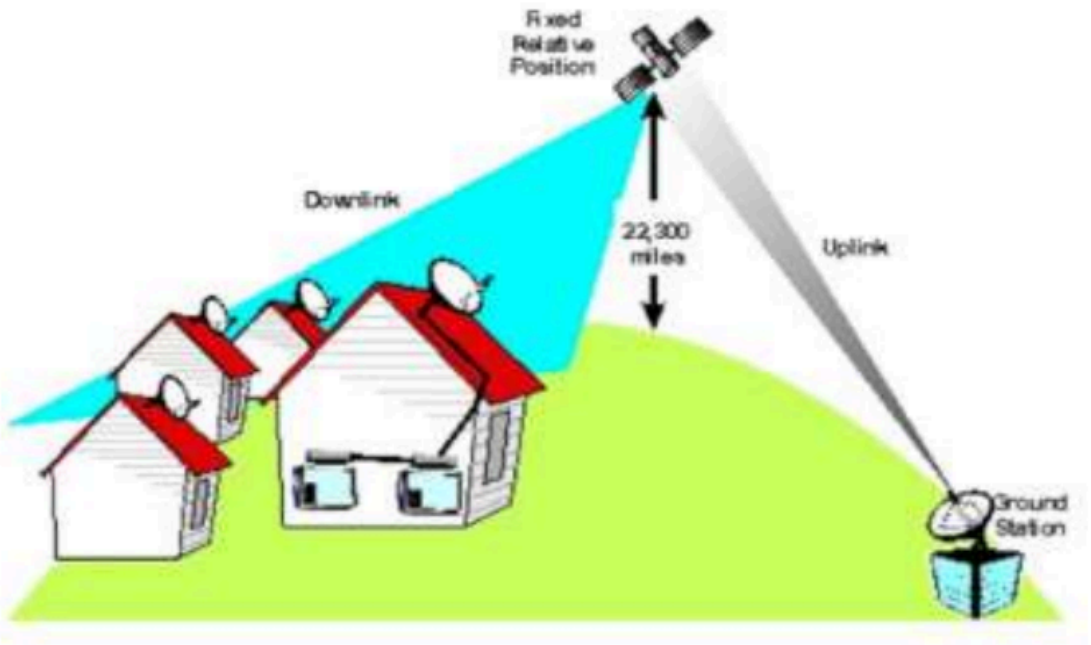
Differently from the PON analyzed before, if a user arrives or becomes inactive, there is no problem: just add or remove a packet from the EPON flow. So there are no slots to manage when adding or removing users.

Wireless and Mobile Networks

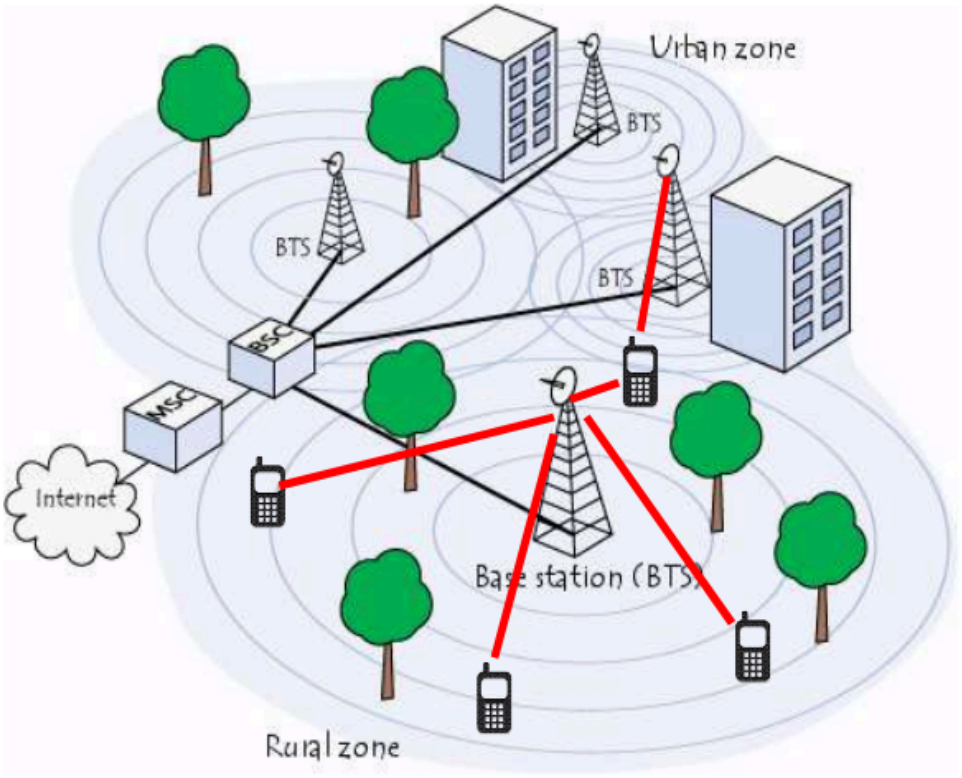
4/12, 11/12, 15/12

TV Systems vs Cellular Systems:

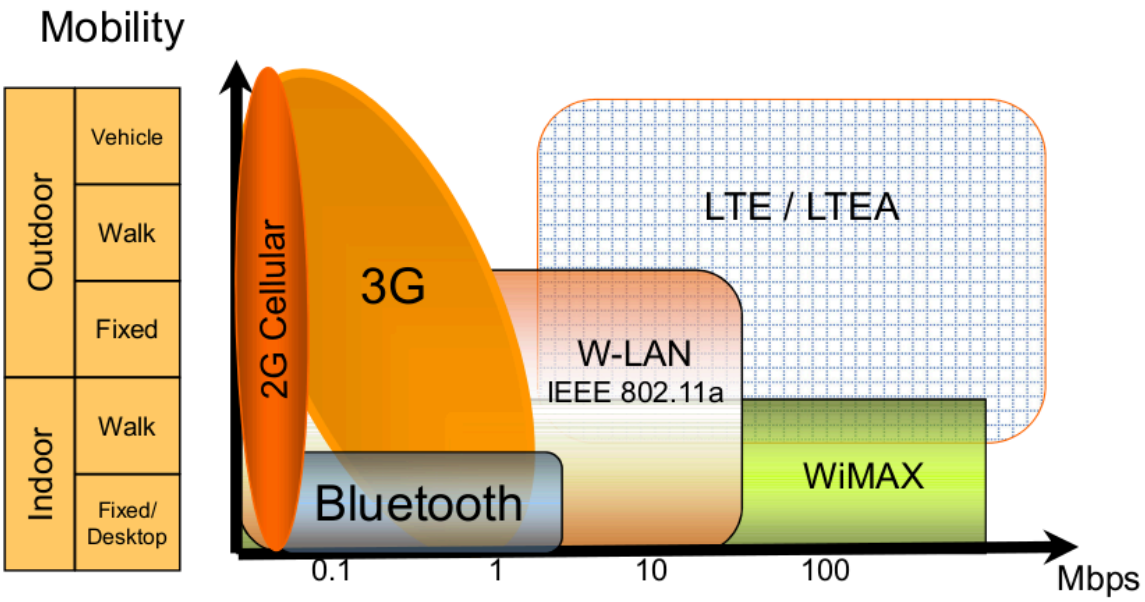
TV Systems have a 1 to n single direction communication: with a satellite or a terrestrial antenna I transmit one unique signal that arrives at the users



Cellular Systems have a n to n double direction communication



Mobility of different technologies
Indoor = LAN
Outdoor = WAN



Fixed Indoor Mobility allows you to move in the home and support *nomadicity:* I can use my laptop in my home, then I can close my laptop and go to another home, not having mobility during the movement but, once I am there I can reconnect again.

Fixed Outdoor Mobility isn't really mobility. It uses FWA: I have a house and a building, on top of the building I have an antenna and the antenna gives me connection since I have an antenna in my house.

Bluetooth is almost fixed. Bluetooth has a very short coverage range, about 100 meters.

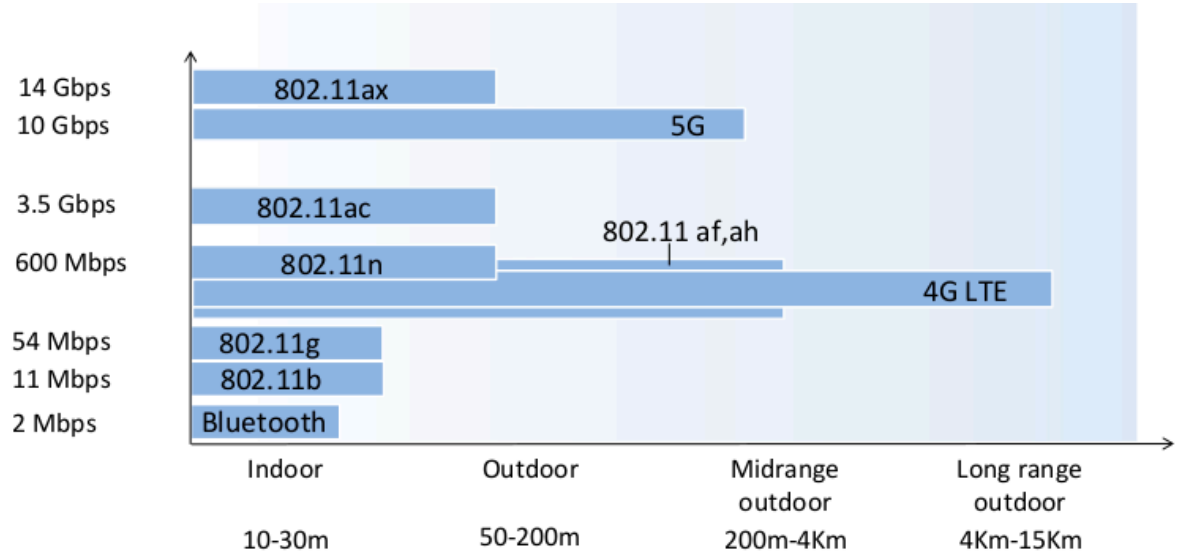
Wi-Fi has similar behavior to Bluetooth with higher bit-rate and capability of supporting mobility. Wi-Fi has about 200-300 meters of coverage.

Wi-Max, evolution of Wi-Fi, for covering higher ranges, in order of kilometers. But unfortunately the standard is not used.

2G is very mature in supporting mobility. The 2G was mainly designed to support voice calls, so you are able to keep a voice call even if you move very fast.

3G and subsequent technologies, since they have to support mobility for data and not just voice calls, are not as mature as 2G.

As shown in the graph below, we have different ranges and data-rates for different technologies.



This is because we have to consider

1. Frequency: the lower is the frequency (so the larger the wavelength) the longer is the range. So with 5G, for instance, the wavelength will be shorter and so we will need a shorter link with "antennas".
2. Power: the higher the power the longer the distance. So with 5G, for instance, we can play with power to lengthen the link.

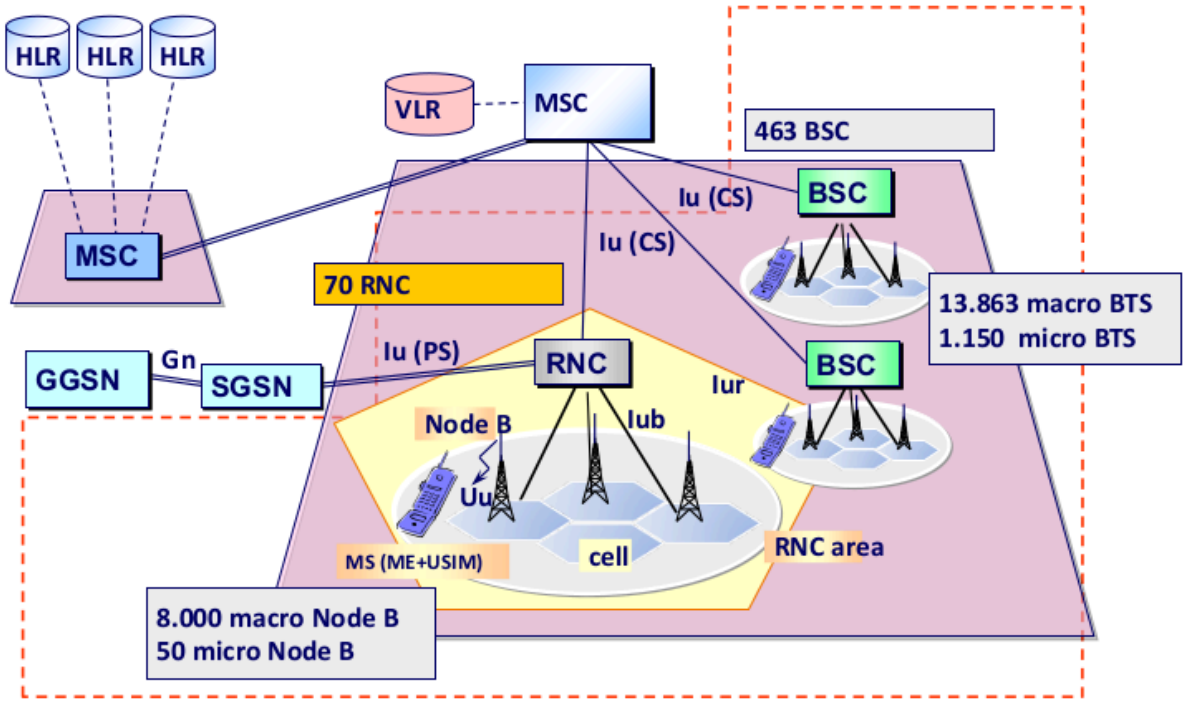
Example of Cellular System Architecture (2G and 3G)

Has the following advantages: mobile and on-the-go connectivity. Used in smartphones and mobile data in general. We are now in the fifth generation of cellular access (5G). In the cellular network we have some servers (square) that have network function, so they do computing. Precisely, they manage the mobility and the access to the resources. Then we have the antennas interconnected to the servers and the servers are connected to the routers and the rest of the network. So, the cellular network has a part that is wireless and a part that is wired (connecting antennas, servers and other part of the network). The wireless part is few compared to the wired part, and the wired part nowadays is most fiber. 5G Access: ultra-fast and low latency connectivity

3G is also for data transmission but the main focus remains voice calls
first time data transmission used but not so much as in 4G
so not only voice calls as in 2G, also data transmission but mostly voice calls

We cannot have contemporary transmissions of H1-AP_A and H2-AP_B since H1 and H2 are hidden terminals.
We cannot have contemporary transmissions of H1-AP_A and H2-AP_B since H1 and H2 are hidden terminals.
4G is for data transmission for the following application: videostreaming

The 1G (First Generation) was analog communication, 2G digital, 3G, 4G we are in the 5G and we will soon have the 6G. This in the picture is the Access Network Architecture for 2G and 3G:

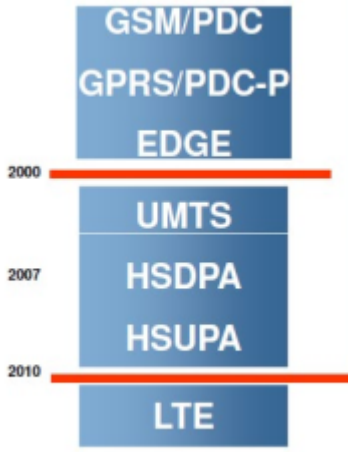


We have to split our wireless link in cells in order to use at the best the radio/wireless spectrum. In a cell all the elements using the spectrum are using the same portion of the spectrum. The cells are built up in order to not interfere with each other.

To manage this environment we need antennas, called base stations, that are the emitting station for the cell. We have a base station in each cell. Each base station, coordinated with a Radio Network Controller (RNC) has the control of the communication of the cell. So the RNC coordinates multiple base stations. The RNC also interacts with gateways that are the connection of the part of the network shown in the picture (that is the access part) with the network part, the classical networking part.

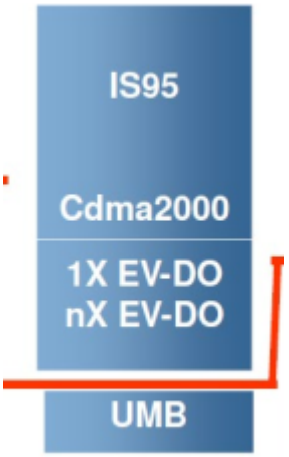
With this environment we are able to communicate wireless and support mobility. The Mobility Service Center (MSC) is the one who manages the mobility: it searches for our position. For instance when we are in roaming we are not in our network operator, but we are using another network

Evolution of Cellular Network:
3GPP is the entity who manages standards regarding cellular networks, such as the GSM GPRS LTE etc.



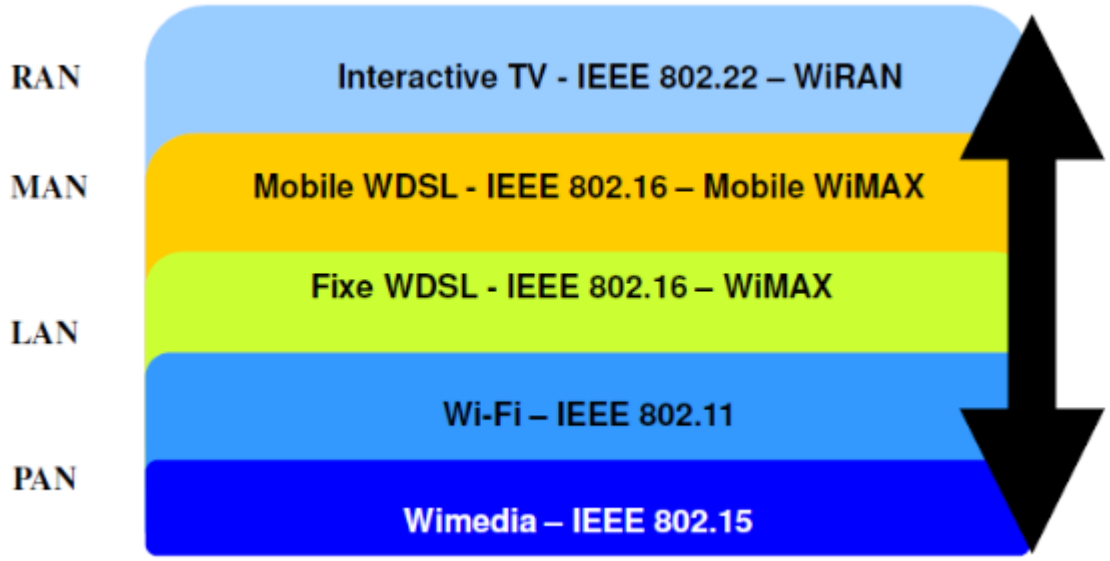
GSM/PDC and the other until 2000 are 2G standards, UMTS and the other until 2010 are 3G standards, LTE is a 4G standard.

Some time ago there was a distinction between cellular networks in Europe and Japan and cellular networks in the US, so the standards were different. The US had Cdma2000.

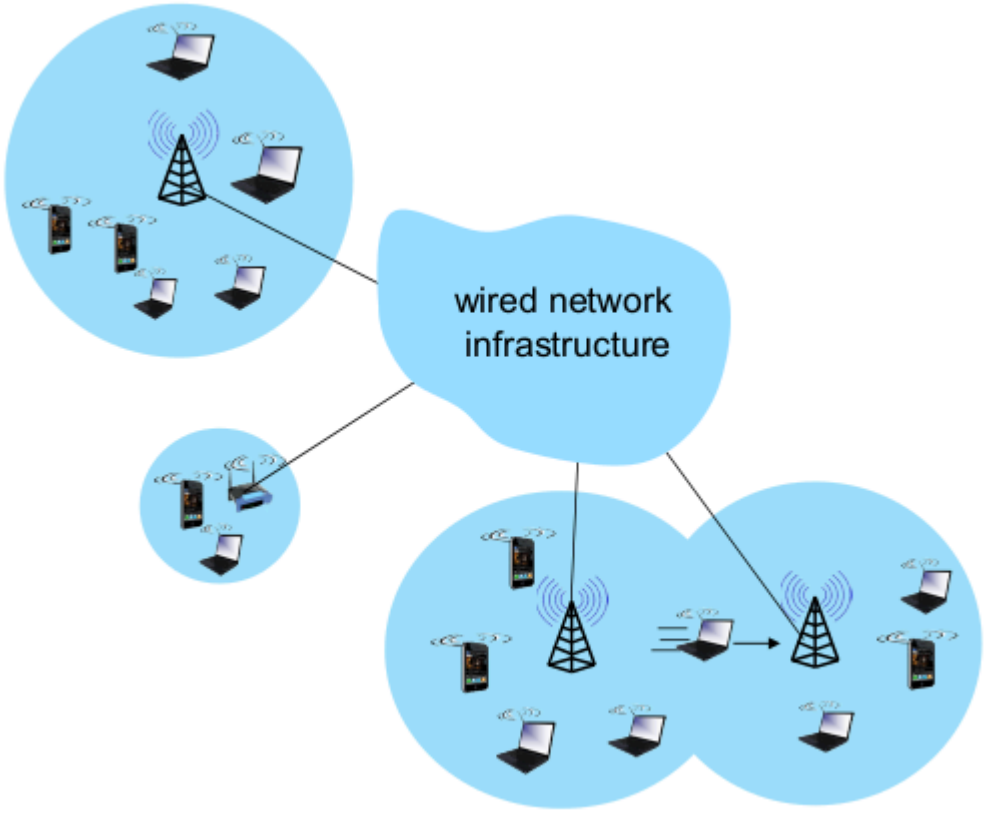


Then there is another framework, the framework of the 802 family. In which we have Wi-Fi

But Wi-Fi provides wireless connectivity at most in the LAN. With WiMAX and WiRAN we extended that in the MAN and the RAN (Regional Area Network). WiRAN used another spectrum, not the one of the Wi-Fi. It transmits on top of the television broadcasting interface. WiMAX was quite successful, so we began implementing mobility on it, since Wi-Fi by default does not support mobility.



Wireless Network Architecture with Infrastructure:
This is the architecture of a Wireless Network



There are wireless areas covered by an antenna, a real network element that provides the signal to the wireless-hosts (PCs, smartphone, IoT, ...), named User Equipment (UE). The antenna is called “base station”, “node B”, “eNB” in 4G and “gNB” in 5G.

The antennas are interconnected to the core network (Wired Network Infrastructure or Backhaul), through a wired link in fiber. This is the backbone of the cellular infrastructure[13].

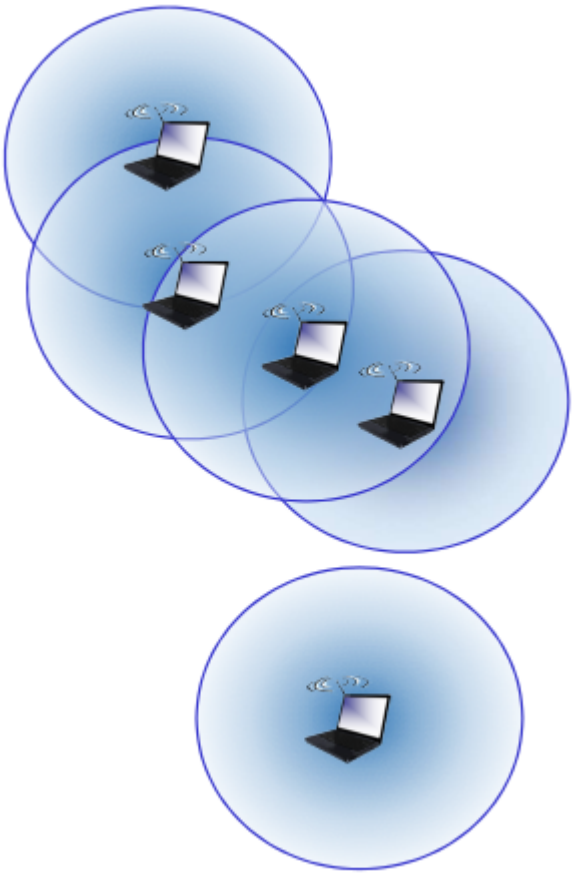
There is an important exchange of information mostly to control the network in the Wired Network Infrastructure.

This architecture supports both *single-hop* (a host is connected to a base station that connects to the Internet) and *multi-hop* (a host has to relay through several wireless nodes to connect to the Internet, and this is called Wireless Mesh Network).

Ad-Hoc Network Architecture (no Infrastructure)

We can also have wireless networks that don't use any infrastructure or antennas at all and they are called *ad-hoc networks*. In this case there is not connection to larger internet.

[13] The real backbone is the internet, but in between the access and the core there is this part. Is present more or less in all wireless infrastructure, in wi-fi is very light, here it is very complex.



Let's make an example: if I interconnect my laptop with the one of Prof. Cuomo this is a little wireless network, then Cuomo can connect to another student and so on. These networks are fully dynamic, opportunistic (so I can use the laptop that is more comfortable each time) and support nomadicity (we can set up the network where and when we want).

Also this type of network supports both *single-hop* and *multi-hop*. An example of a single-hop ad-hoc network is the hotspot, whereas IoT is multi-hop.

This type of network have 2 interesting applications:

1. Vehicular Ad-hoc NETwork (VANET). E.g. you want to inform the other neighbor cars that your brakes are broken or information about the traffic.
2. Internet of Things. Small things interconnected one with the other. They are not User Equipment anymore, since they are low cost, low power, low capacity and so they don't have the power to be interconnected to a base station. IoT uses the architecture in a multi-hop fashion. Example: railway monitoring, small sensors every 10 meters let's say. They cannot interconnect to the cellular network so they do multi-hop transmission and after this length of interconnection you have the main information sent to a base station.

Criticality of Wireless Link:

- Path Loss: decreasing in signal strength. Given an antenna, the power decreases as a function of the distance, so the smartphone receives less power. This means that we can't transmit too far because of power loss. Path loss depends also on the frequency that carries the signal

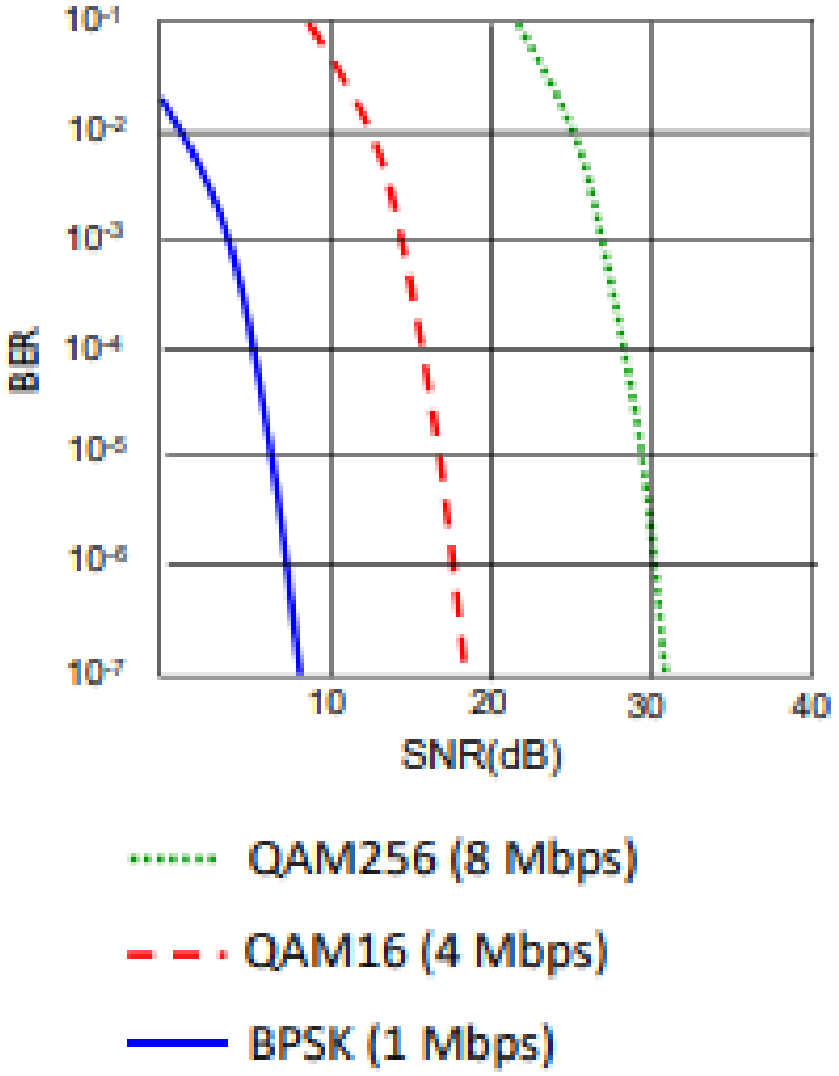
- Interference from other sources, if we have multiple devices in the same frequency bands. SINR (Signal + Interference Noise Ratio) measures the performance in the communication.

  $SINR = P_T / (P_I + P_N)$

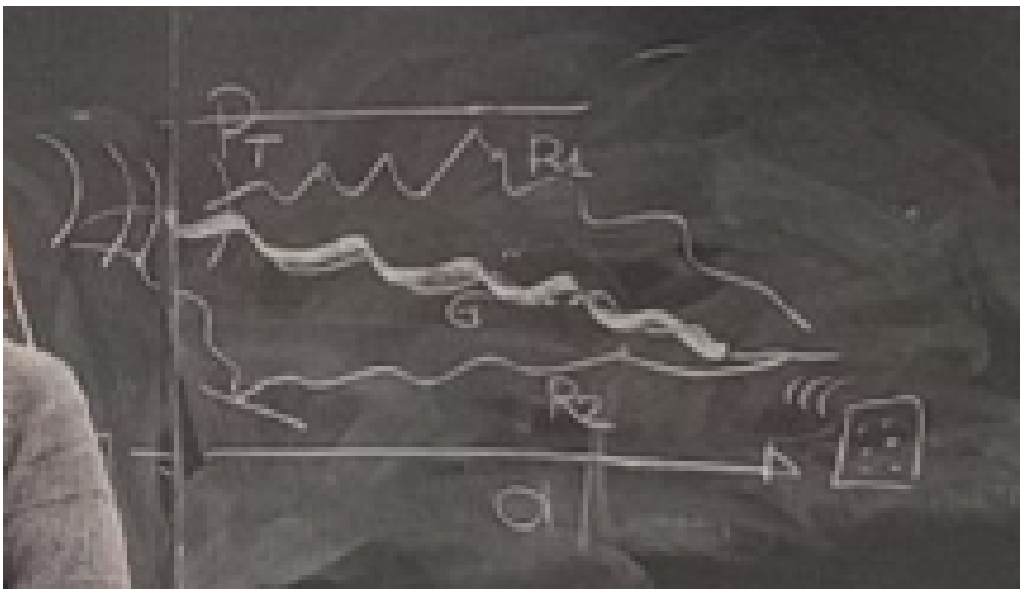
  if the power of interference is higher is not good, and obviously also if we have high noise. If P_T received is very lower from P_T transmitted our SINR goes down. We can have interference also with devices that are not for communication (e.g. machines).

  If SNR is low BER[14] is high.

[14] BER is how much we fail when we try to reconstruct the signal.



- Multipath propagation: if in the space of transmission there are multiple surfaces these surfaces may reflect the signal. e.g. the receiver receive 3 replicas of the signal in slightly different times (if we have 2 surfaces)

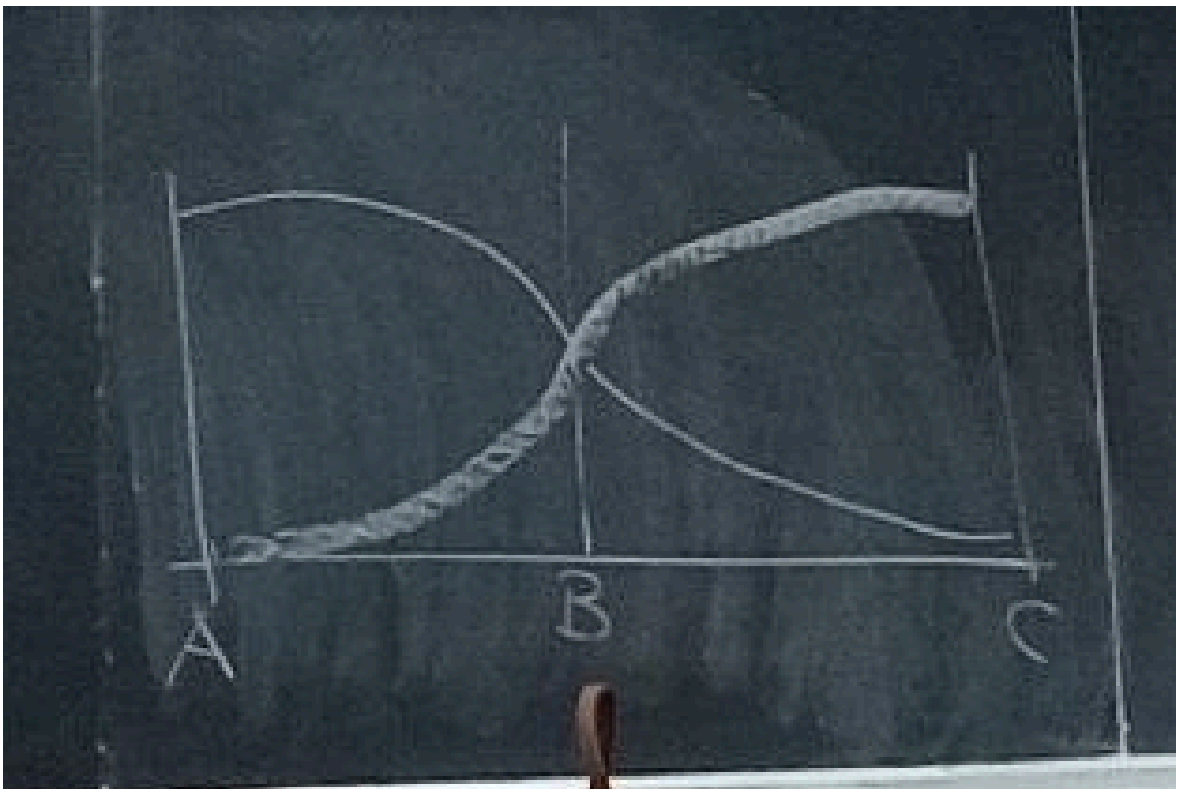


This is not good unless you are able to recognize which are the replicas. If I am able to recognize how this eco (the replica) happens I can synchronize it with my original signal to boost the signal, so to transform a problem into a benefit. To do that you have to model the environment and the critical paths that a signal can take.

The higher the frequency the easier it is to get reflected by small things. E.g. leaves of trees or drops of rain may reflect signals.

Hidden Terminal Problem:

A is transmitting to B and C to B. A is in radio visibility with B and B with C. C and A are not in radio visibility one with the other. For instance: A and C can be two mobile hosts of Wi-Fi and B is the Access Point. When A transmits, due to pathloss the signal decreases. Both signals arrived quite good at B though.

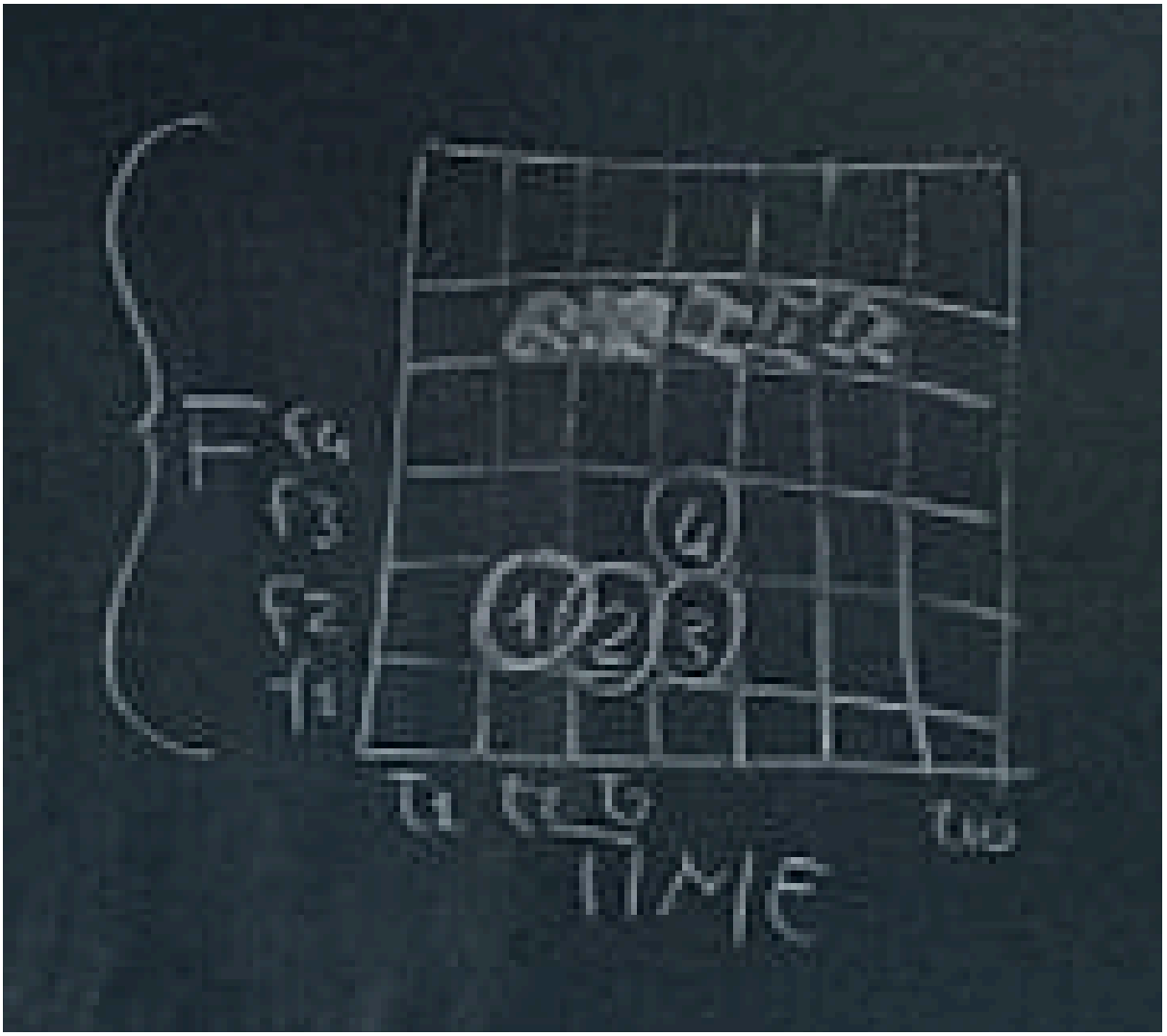


This Is a problem in wireless since if they are transmitting in the same frequencies and simultaneously in time they don't see the communication of each other and there is a collision (this could not happen in wired since each node sees all other ones). To avoid the interference we can use:

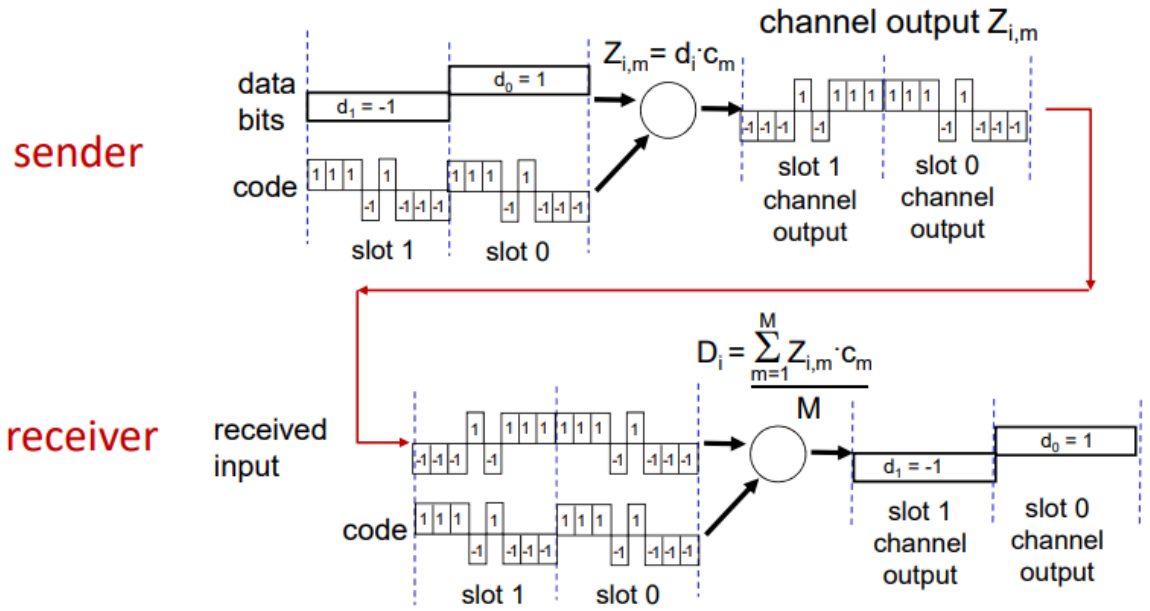
1. Frequency Division. Wi-Fi does this, but we have the problem of a limited number of users and we have to have a bit of padding between the frequencies given to a node and the one given to another. And you have to have B that is able to communicate contemporary in different frequencies since A and C communicates now in different frequencies
2. Time Division: we need someone in the system, a manager, who assigned the time.

Cellular Technologies uses both: in the LTE the spectrum is divided in Time and Frequency. So it is a sort of matrix time slots x sub-frequencies channels

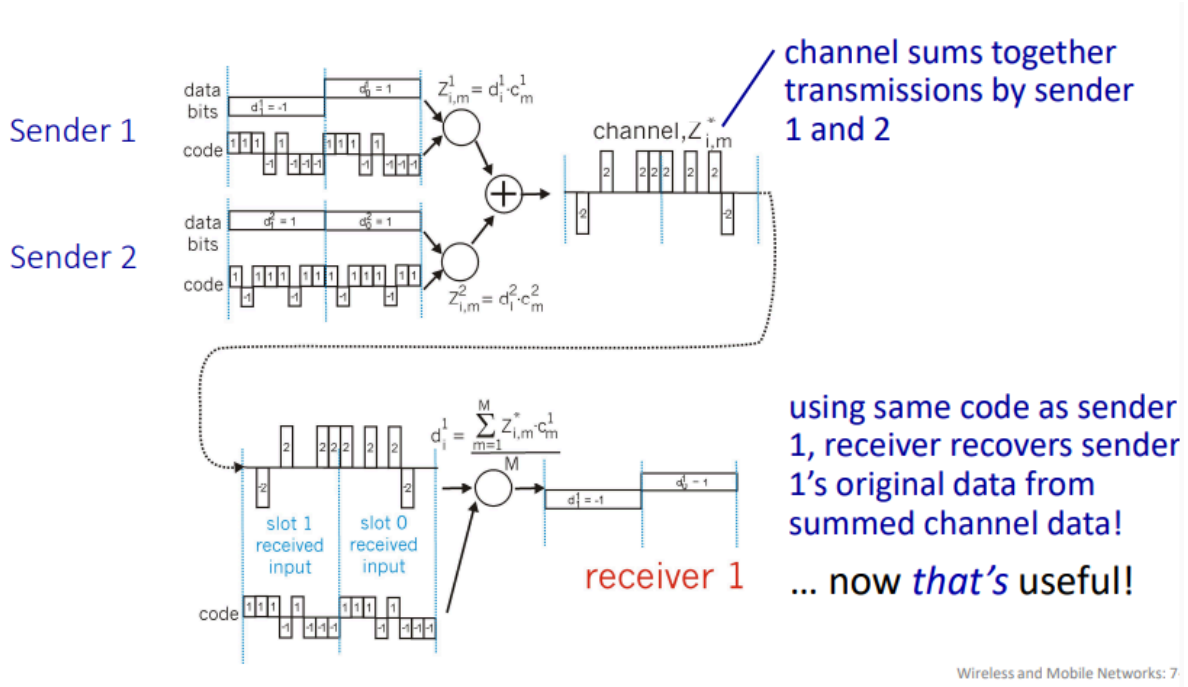
The manager in an LTE network is able to "fill" the matrix for each user. A user can have multiple cells in that.



There is a third possibility, Code Division Multiple Access (***CDMA***): we can transmit at the same time and with the same frequencies. Each transmission is equipped with a unique code. The codes are orthogonal or pseudo-orthogonal:



… but this isn't really useful yet!



The sender transmits the data and the unique code and then the receiver is able to recover the signal. CDMA2000, the USA standard, uses this. While 3G 4G and 5G uses the LTE matrix.

1. Advantages: you can avoid the complexity, since you don't need the LTE matrix
2. Disadvantages: the more the user the longer is the code since we have to distinguish them. Spread Spectrum Signal: the signal occupies a larger bandwidth, so you are generating a signal that requires a big bit-rate and the capacity could be not enough.

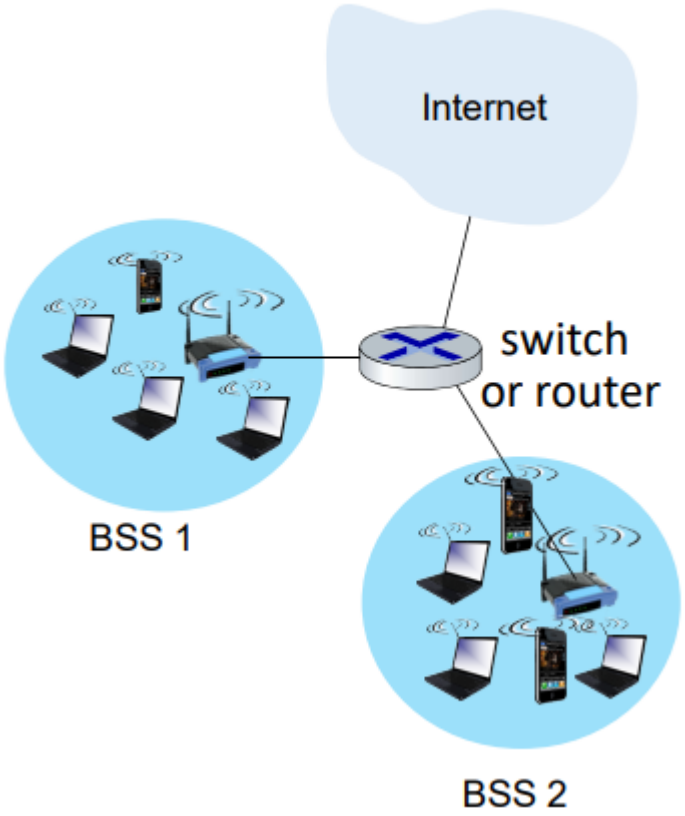
FDMA, TDMA and CDMA are complex, you need some intelligence. If you want to make it simple and distribute the intelligence we can use CSMA/CA, which is used in Wi-Fi.

Wi-Fi: IEEE 802.11 Wireless LAN

This family of standards uses 2.4 Ghz or 5 Ghz spectrum. 802.11af and 802.11ah are using a different and lower spectrum since the lower the frequency the longer the distance.

IEEE 802.11 standard	Year	Max data rate	Range	Frequency
802.11b	1999	11 Mbps	30 m	2.4 Ghz
802.11g	2003	54 Mbps	30m	2.4 Ghz
802.11n (WiFi 4)	2009	600	70m	2.4, 5 Ghz
802.11ac (WiFi 5)	2013	3.47Gpbs	70m	5 Ghz
802.11ax (WiFi 6)	2020 (exp.)	14 Gbps	70m	2.4, 5 Ghz
802.11af	2014	35 – 560 Mbps	1 Km	unused TV bands (54-790 MHz)
802.11ah	2017	347Mbps	1 Km	900 Mhz

The architecture of a 802.11 LAN is the following:



A Basic Service Set (BSS) has wireless hosts communicating with a base station (Access Point). Each BSS is interconnected to a router or a switch and the router is the interface through the internet. In our house we have a single BSS connected to a router. The access

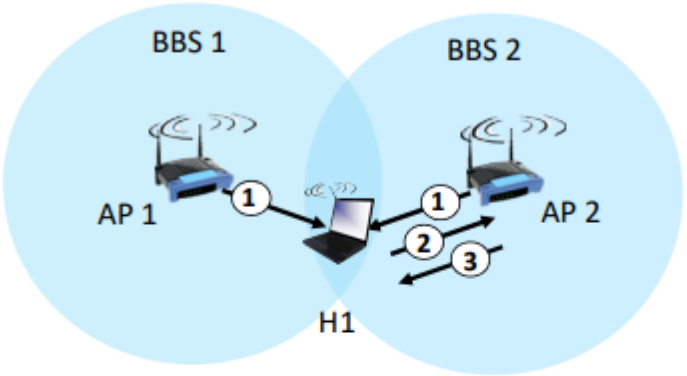
point and the router may be the same object. The spectrum is divided into channels and an AP admin assigns frequencies to the APs. If I put an AP in this room operating on a sub-channel and another in another room with another sub-channel, we can operate contemporary since there is frequency division. It is still possible to have interference since neighbors APs can have the same channel assigned.

When a mobile enters it should associate with the AP. This association happens by selecting the AP in the area, with these passages:

1. The host scans the channels, with passive or active scanning
2. After found the APs, it chooses one to associate with
3. The host may perform authentication
4. A DHCP server gives an IP address to the host in the AP's subnet

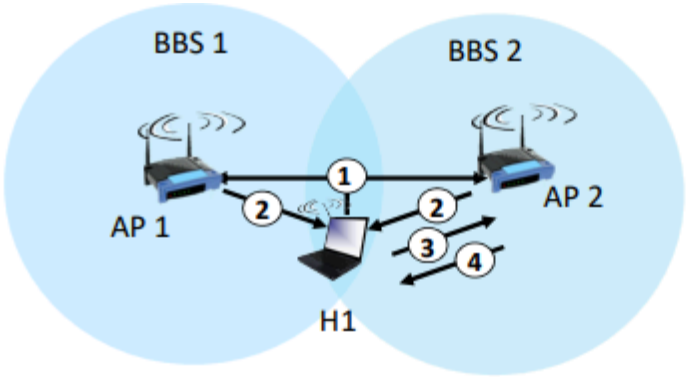
There are two possibilities:

1. Passive Scanning: H1 scans the air, and the AP sends broadcast signals called beacon frames in their coverage areas. If there are more APs H1 receives a beacon frame of both and selects the best, in terms of highness of received signal. H1 then sends an association request and the AP responds with an association response.



passive scanning:
(1) beacon frames sent from APs
(2) association Request frame sent: H1 to selected AP
(3) association Response frame sent from selected AP to H1

2. Active Scanning: H1 spreads the request (probe request) to known APs. The APs send a probe response and so H1 sends an association request followed by an association response.



active scanning:

(1) Probe Request frame broadcast from H1
(2) Probe Response frames sent from APs
(3) Association Request frame sent: H1 to selected AP
(4) Association Response frame sent from selected AP to H1

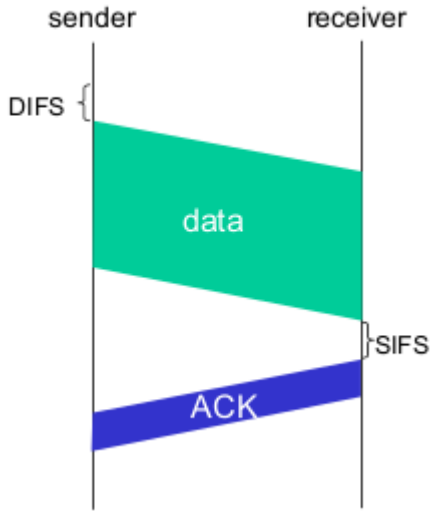
Both are often implemented: when you enter your house you do an active scanning (is there my Wi-Fi in which I always connect?). On the contrary, if you go to a bar you have to scan the system.

Active scanning is more secure since I know the ID. On the contrary though, in active scanning, when transmitting the probe request, H1 sends a list of well known Wi-Fi, so I am listing to possible attackers a list of Wi-Fi I met, so you can know where I went, bar, concerts and so on… so a problem of privacy.

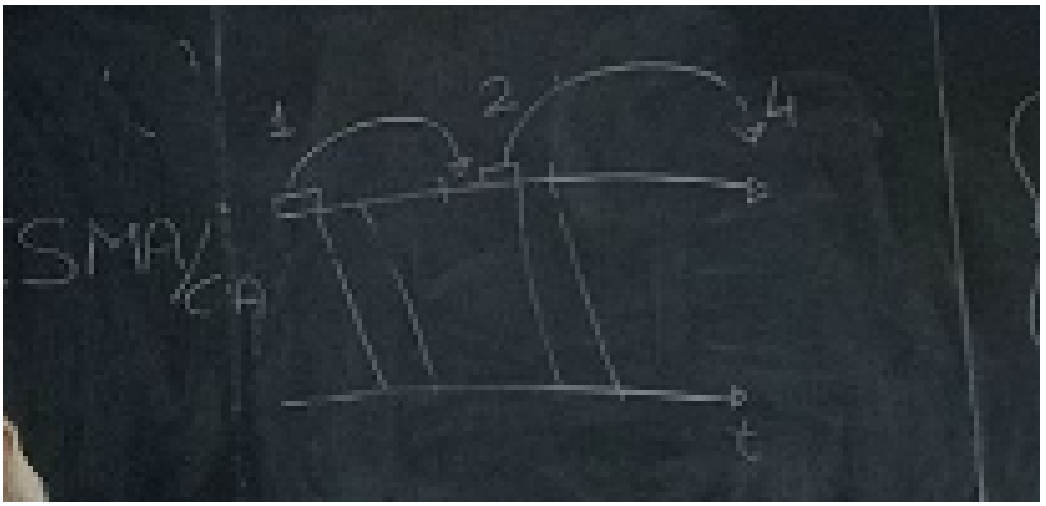
CSMA/CA for Hidden Terminal Problem:

Hidden terminal problem is solved in a simpler way than FDMA, TDMA and CDMA. We use CSMA/CA (Carrier Sensing Multiple Access Collision Avoidance)[15].

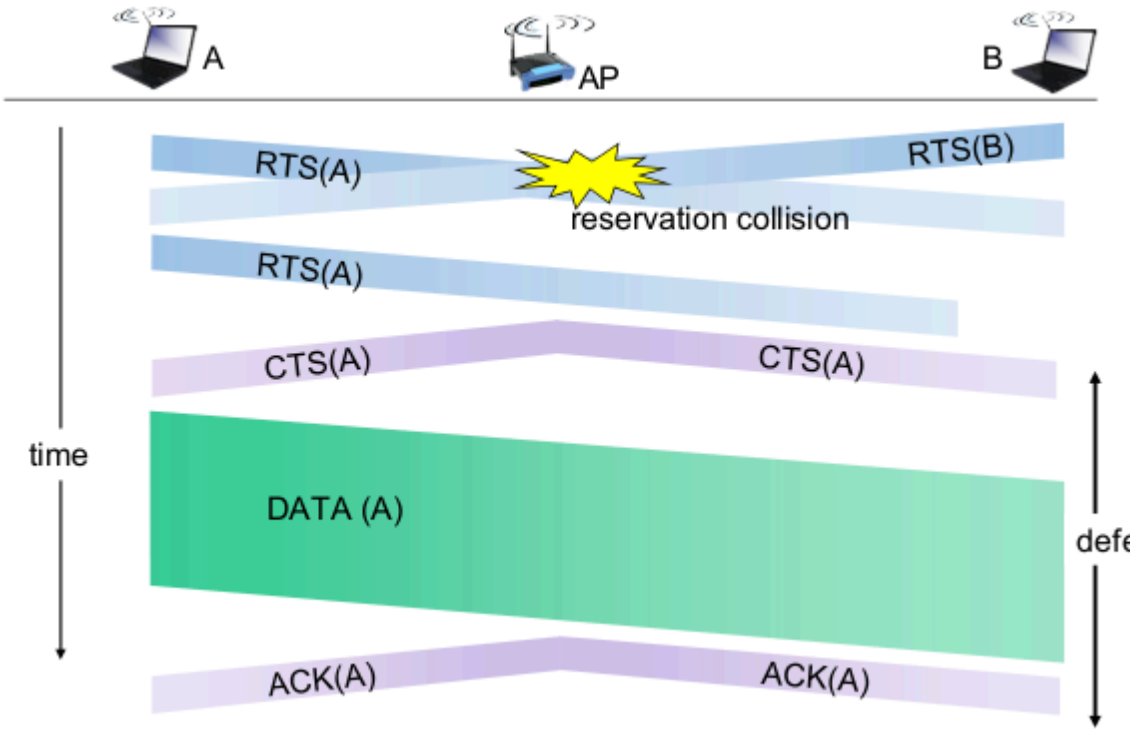
[15] No Collision Detection (CSMA/CD), as in Ethernet, since A receives weak signal from B and can't sense the collision, but only the channel.



The sender has to listen to the channel for a period of time called DIFS. If you during the DIFS find out that the channel is free you are able to transmit. Since you are not sure that your data does not collide, the receiver must transmit back to you an ACK. B transmits the ACK after the SIFS (shorter than DIFS). What happens if during the DIFS I see that the channel is occupied? I postpone my transmission in time starting a timer (random backoff time) that is k * RTT with k in [0, 2^n-1] where n it the number of attempts. So the more you attempt the more you will wait. I don't receive the ACK. I have to transmit again increasing the random backoff time.



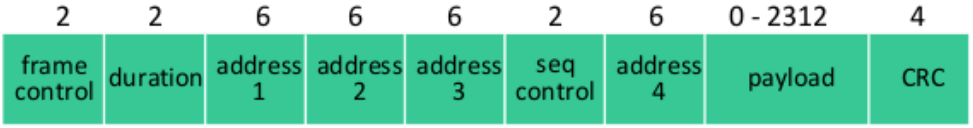
Another way of doing collision avoidance is to book the channel:



I transmit a small packet, the RTS. Once this packet arrives to my intended receiver the receiver should answer with CTS. I can now transmit my packet and I am sure that nobody else will transmit. B is not aware of the RTS since the message doesn't make it to it. For this reason the CTS is broadcasted. B understands that the channel will be busy. So B will stop and eventually go to sleep to save power, whereas A can transmit. After the transmission A will receive the ACK. In the RTS and CTS there is also a time: I would like to transmit the data for e.g. 2 minutes. In this way it is all distributed and there is no centralized manager.

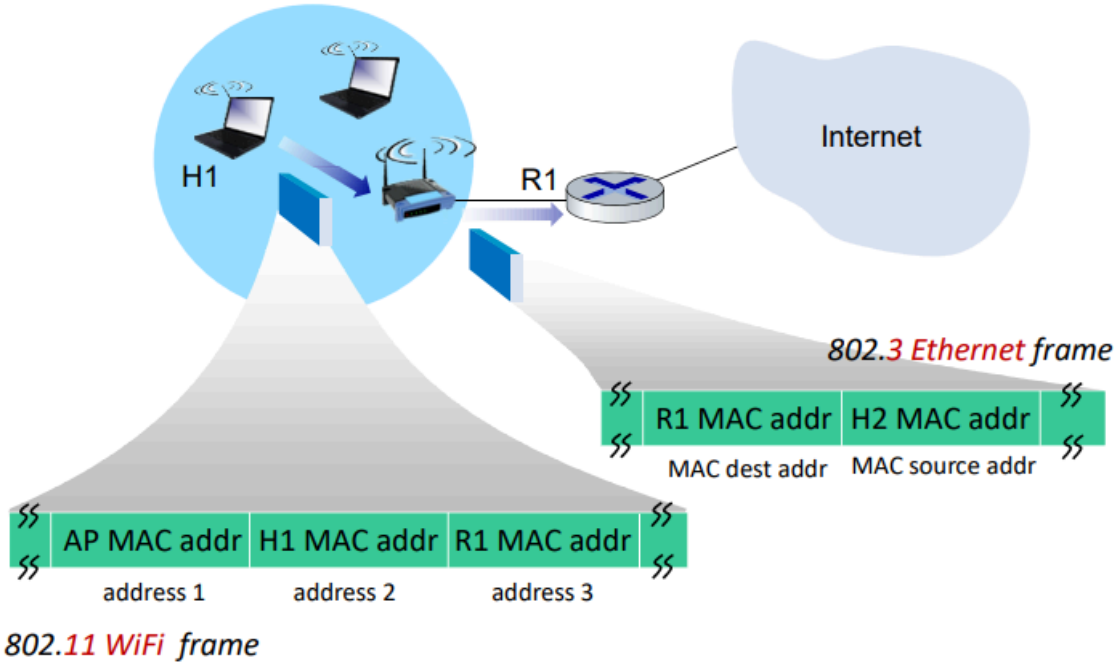
As shown in the picture, the RTSs can collide. In that case The AP will not transmit the CTS and the two will reschedule the transmission at different times.

802.11 Frame:



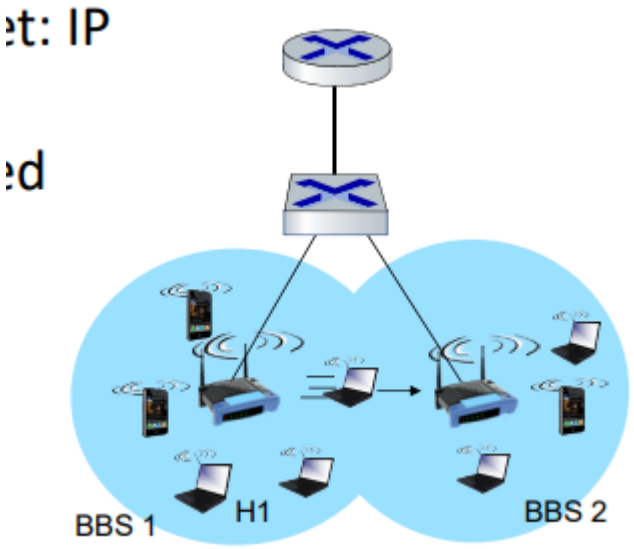
1. Duration: duration of reserved transmission time (RTS/CTS)
2. Address 1: MAC Address of wireless host or AP to receive this frame
3. Address 2: MAC Address of wireless host or AP transmitting this frame
4. Address 3: MAC Address of router interface to which AP is attached
5. Seq Control: frame sequence number for reliable data transfer
6. Address 4: used in ad-hoc mode

There is the address of the access point, the router, and the host I would like to transmit the data to:



The router should distinguish between user that are in a BSS, that should be sent in the internet and so on

802.11 Mobility:
A host can move from a BSS to another keeping its IP address if the two BSS are in the same IP subnet (same infrastructure).



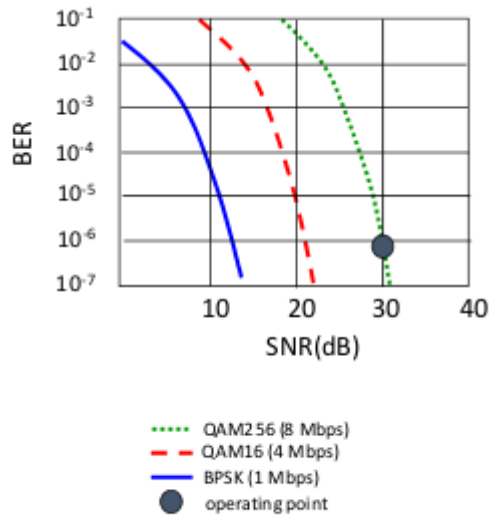
In the case above the switch, seeing from which port the packet of H1 are coming, can “remember” the position of the host and know where to reach the host if packets with destination H1 come.

If the host moves to another infrastructure there is no connection in between the two infrastructure and the host has to associate to a new AP with a different IP address.

Mobility of Wi-Fi is low mobility, we can't move too fast or the system will not manage the movement of the user properly

In the cellular network the mobility is mature since I can move from an area to another keeping the connection.

Rate Adaptation:
The transmission rate (so the physical layer modulation technique, QAM256, QAM16…) changes dynamically w.r.t to the distance of the host from the AP. So when the host is going far away from the AP the SNR decreases, the BER increases and when the BER becomes too high the network switches to a lower transmission rate that can provide a lower BER. If we have a higher BER to use at the best our channel we need to use a modulation that is able to exploit at the best this bad condition. We use a modulation that achieve a low bit-rate (e.g. BPSK)



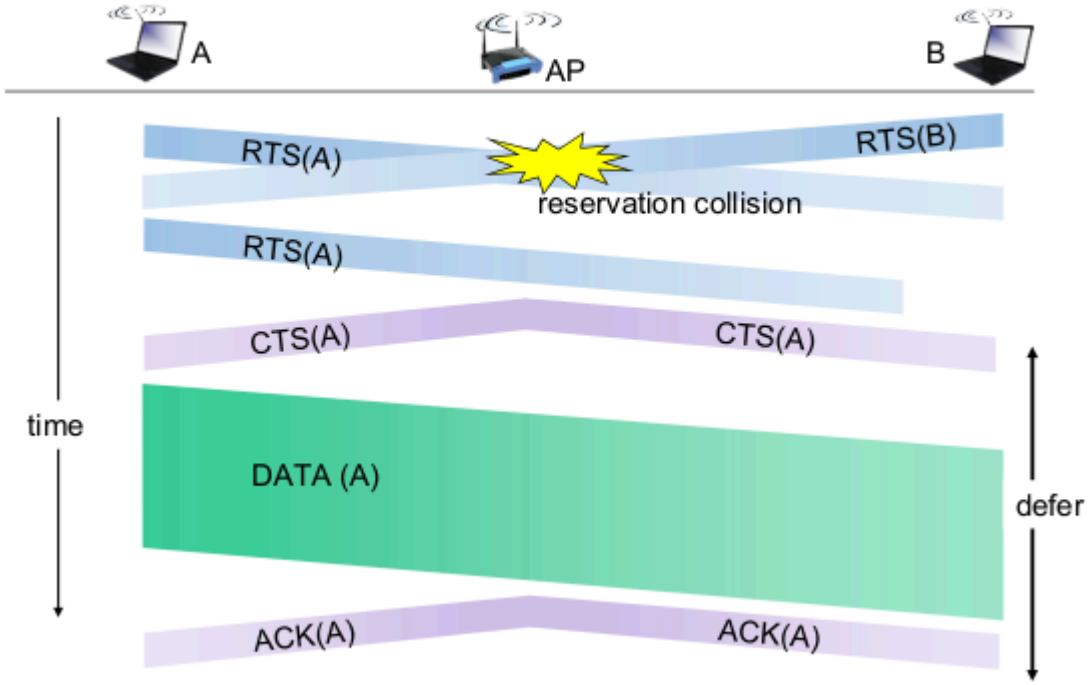
Not all of us will experience the same SNR:

1. we are in different distance from the AP
2. we can have different reflection
3. other aspects

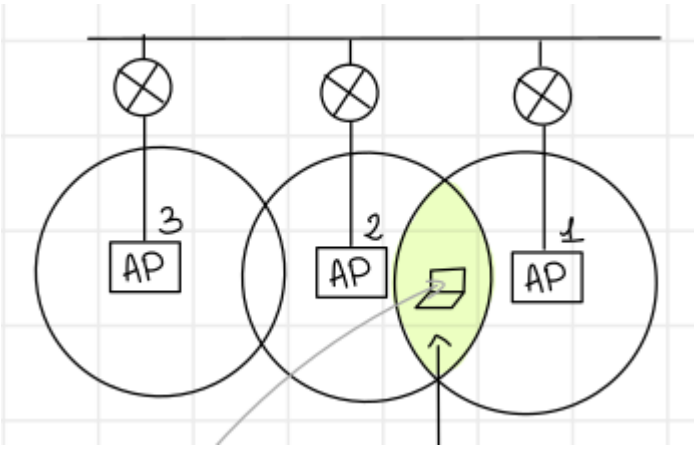
So there is a personalized SNR. The system is smart and for the user who has good SNR provides high data-rate

Wi-Fi Power Management
When you use wi-fi you spend a lot of energy (battery of the smartphone). I need to receive the beacon frame as a host. It is important that I stay alive only when the beacons arrive. E.g. every 10 seconds the host receives the beacon but from one beacon to the other, if the

host is not doing anything, I can go to sleep and postpone its transmission in time and in between go to sleep. In the CTS it is written how much the transmission will last in time (duration field in the frame)

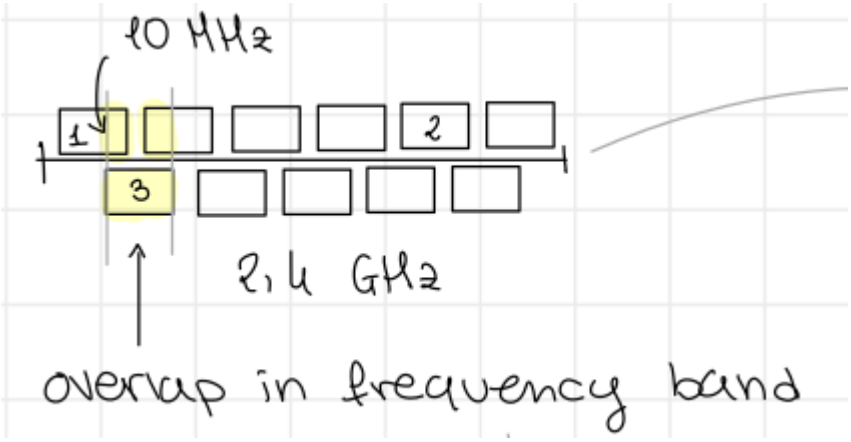


Frequency Overlap:



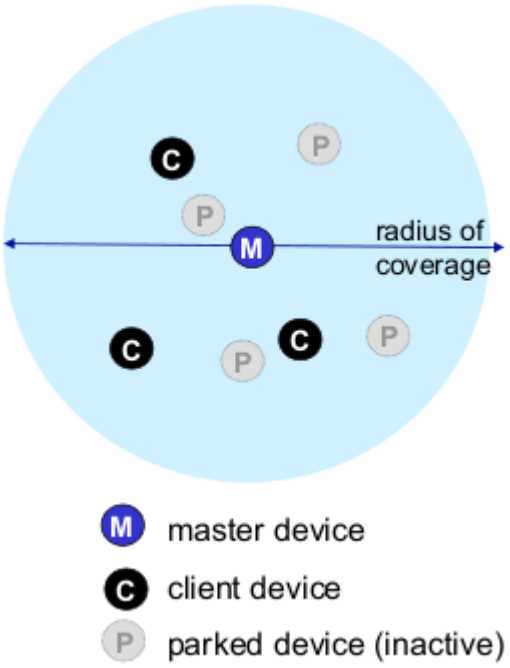
In the same infrastructure, if 2 AP overlap in coverage area, they have to have different subchannels.

The overlap is both in covered area and frequency band

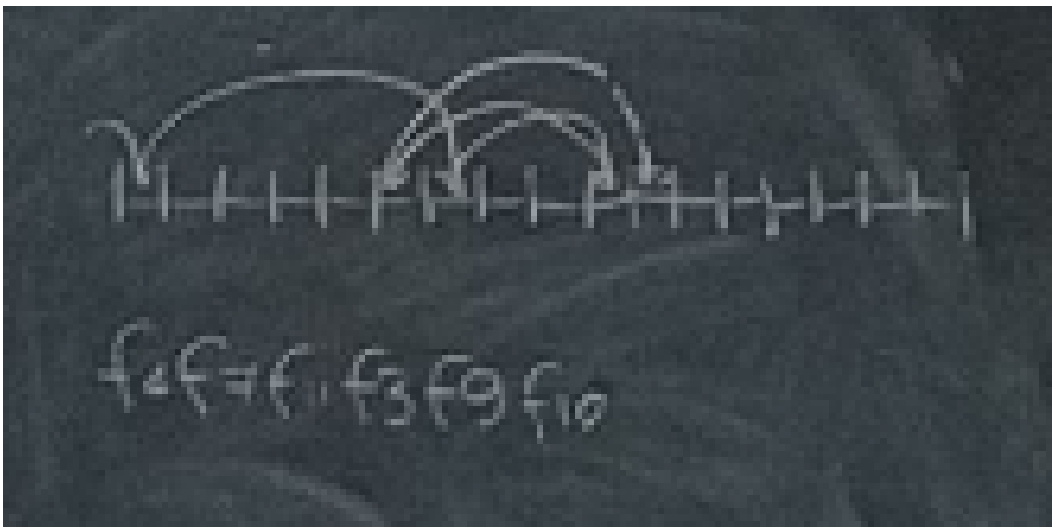


They don't physically separate the subchannel because of the waste of bandwidth so using overlapping is important. In the pictures above there is overlap in freq but not in physical. The AP that intersect should have frequencies that don't overlap. But, even if they are in the same infrastructure, if they don't intersect, they can have the same frequency.

IEEE 802.15: Bluetooth

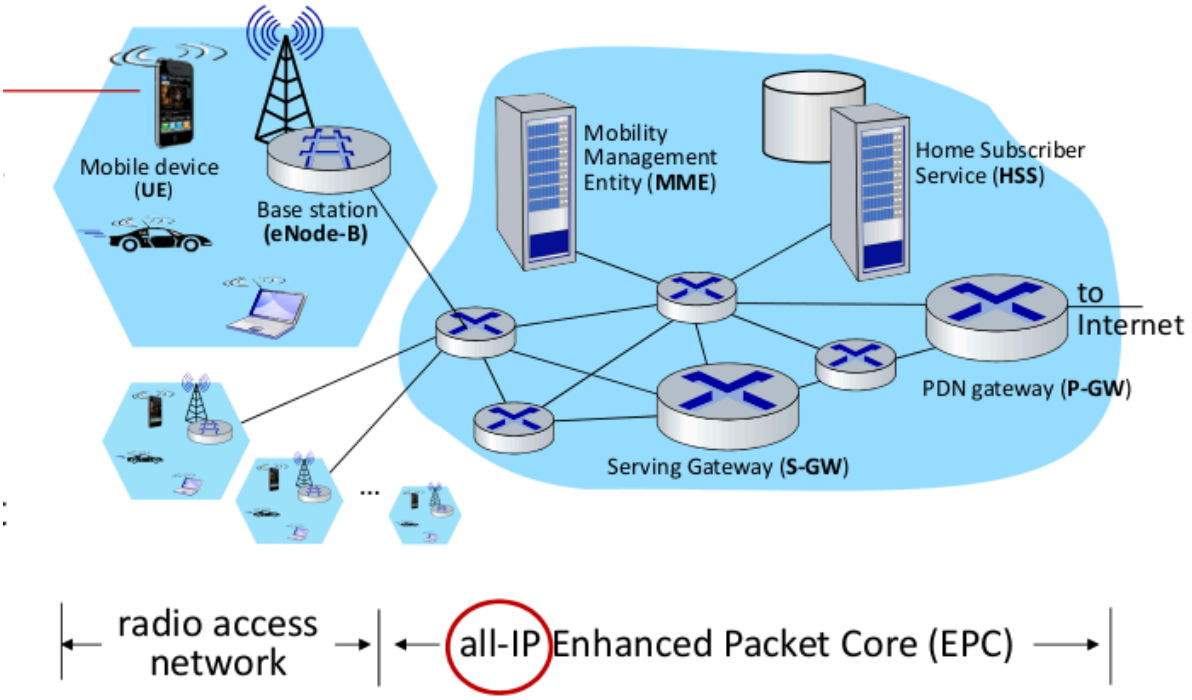


A wireless ad-hoc PAN technology. Bluetooth works in the same spectrum of Wi-Fi. Frequency hopping is used to solve this problem: is a sort of CDMA. We have a lot of small portions of the spectrum, and the unique codes are a list of sub-channels where my transmission will hop.



Since this is CDMA, the receiver should know the frequency hopping code. In this way, even if they are using the same spectrum, Wi-Fi and Bluetooth are using a completely different method to access the media, so Wi-Fi recognizes this frequency hopping as a noise and ignores it.

4G LTE Architecture:

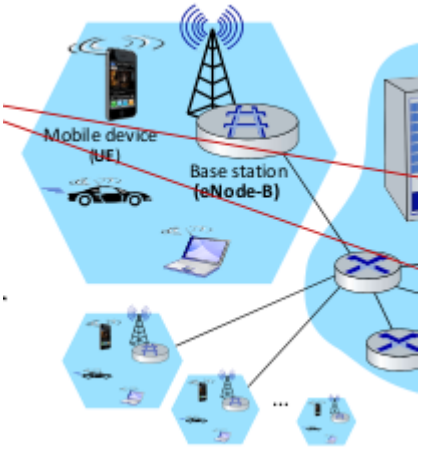


UE: the user that moves in the wireless area

UE is recognized by an identity, the IMSI (International Mobile Subscriber Identity), 64-bit code stored in the SIM (Subscriber Identity Module). The SIM is very secure.

In the peripheral part we have the base station: an antenna that is able to spread the signal and capability to manage the area of the cell.

The RAN is this part:



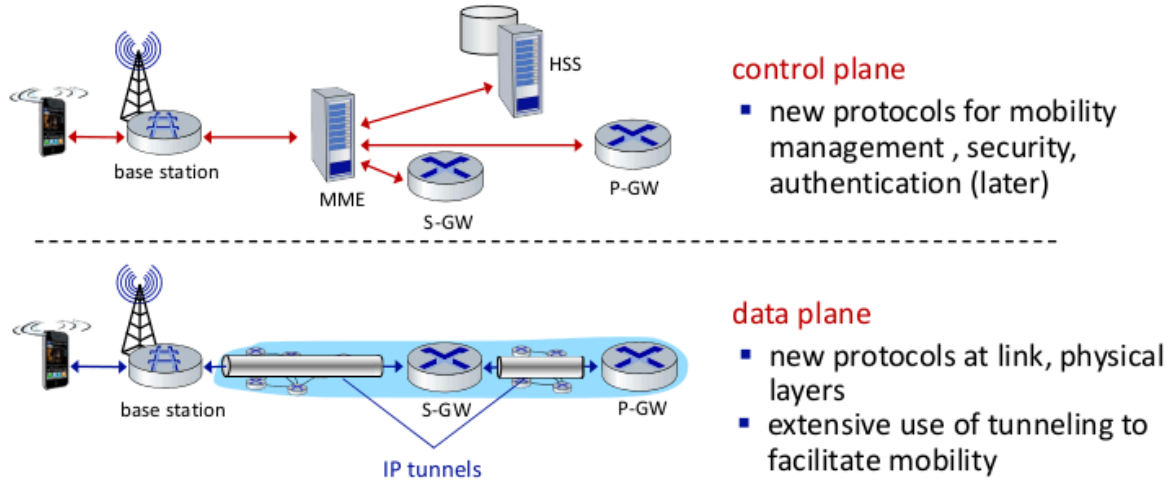
The Identity of the user is stored in a server named HSS (Home Subscriber Service). The SIM should be registered to the network and at the operator side is registered in the HSS. All the operations relevant to the authentication pass through the HSS, otherwise you can't exchange data.

There are some routers: some of them have a high set of capabilities to provide some services. They are S-GW and P-GW. The transmission at the IP level is managed by this internet GW.

MME (Mobility Management Entity): another server that is fundamental in cellular networks because it is where all the mobility aspects are managed.

This is the highest mobility level we know. The MME should interact with HSS to manage the mobility: e.g. a user that moves in an area we should verify that is authenticated.

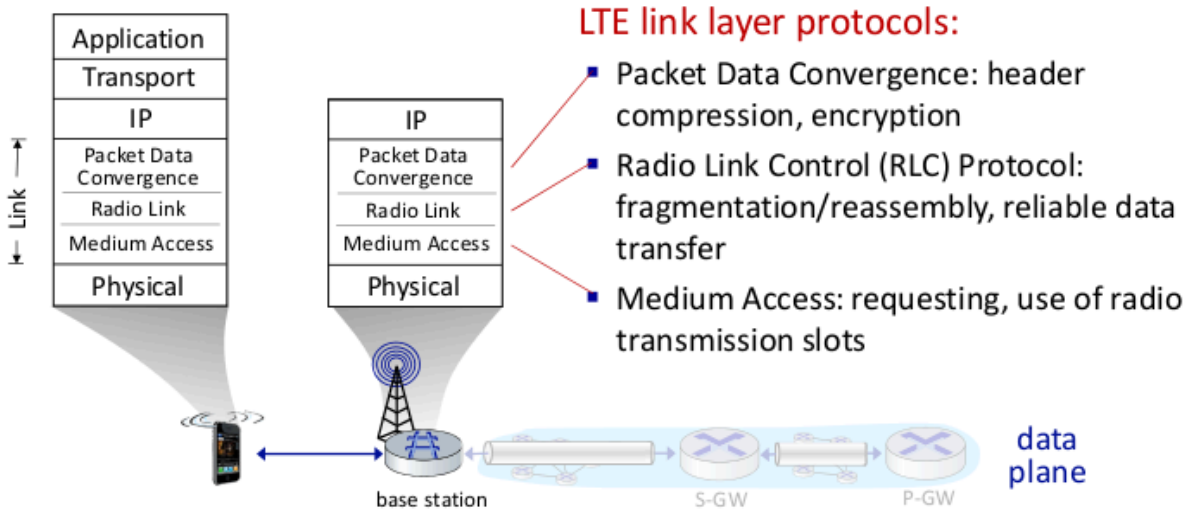
Control Plane and Data Plane:



Control plane: there is an exchange of messages continuously. Mobility management, security and so on

Data plane: exchange of packets happens with IP tunnels.

Protocol Stack in 4G:

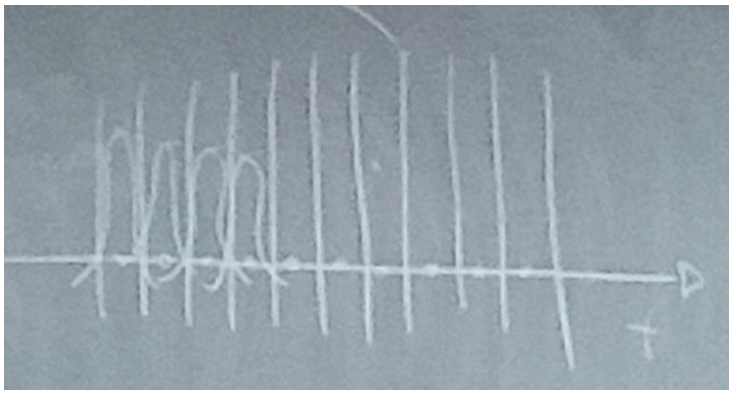


From IP to down is completely different from ISO/OSI. There is a set of protocols able to manage the wireless link:

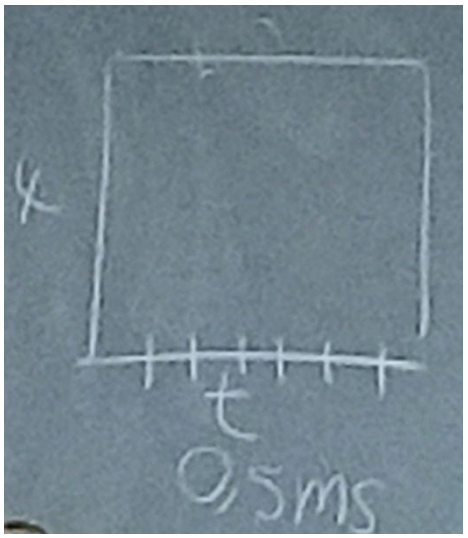
1. Packet Data Convergence PDC: mapping, how to pass from an IP to the radio. Compression, encryption.
2. Radio Link Control RLC: fragmentation, reliable transmission
3. Medium Access MA: requesting, use of radio transmission slots

Orthogonal Frequency-Division Multiplexing (OFDM):

When we have a complex behavior of the channel the best way is to divide the channel in carriers (as in DMT). Differently from xDSL, the way the signal is put is formatted in a way that results orthogonal so the signal in the sub-channel doesn't interfere with the other.

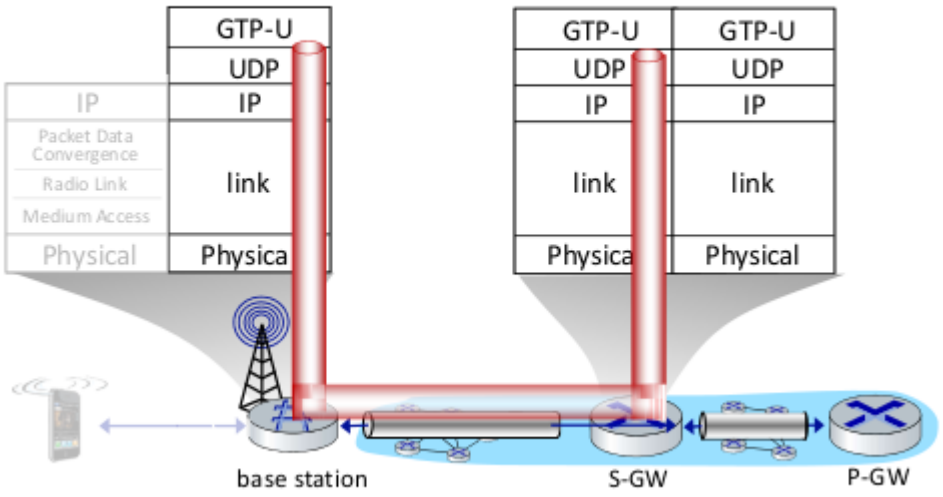


In the frequency-time matrix each time slot is of 0,5 ms

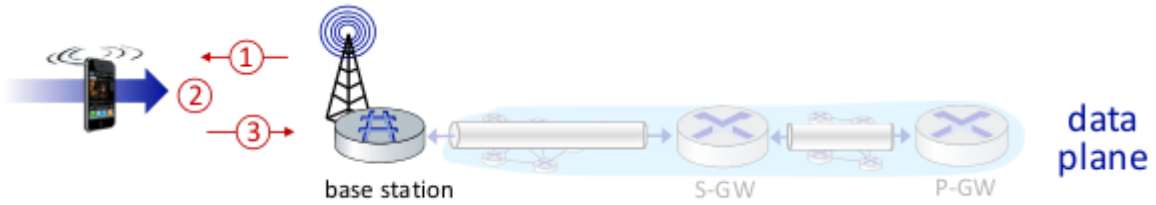


Tunneling:

From the base station the information goes to the GW to arrive at another base station. LTE uses tunneling to provide a connection with IP (a sort of IP session) in which all the info that is taken from the access network is tunneled in the links. To provide this encapsulation there is GTP (GPRS Tunneling Protocol). The data are inserted into UDP segments by using this protocol and UDP uses IP even if the data were not IP because the initial part is not natively IP



Device Association in 4G:



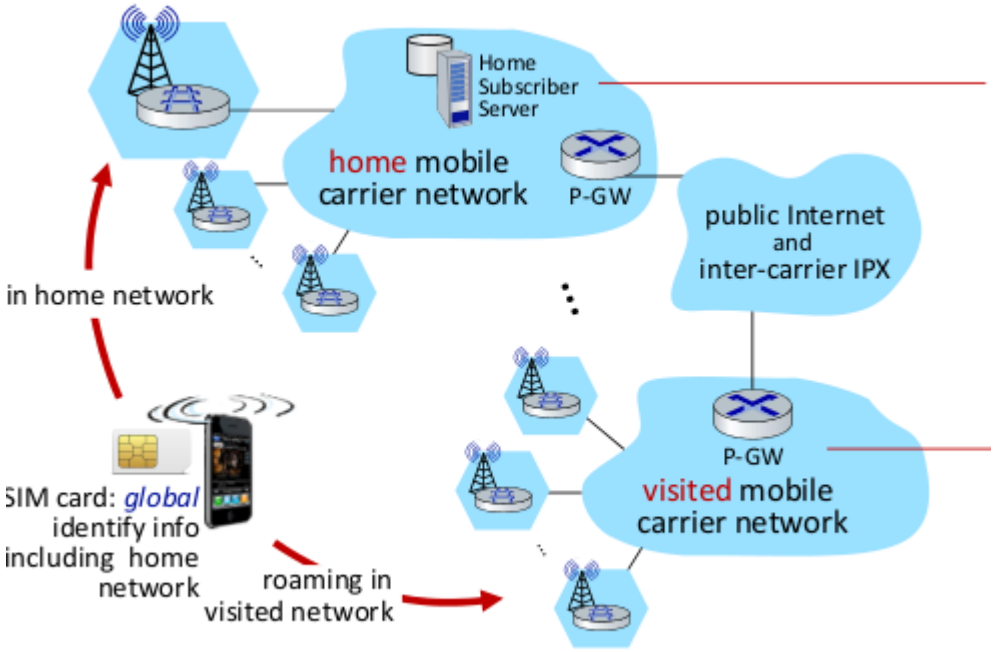
1. BS broadcasts a signal, for instance every 5 milliseconds, on all frequencies.
2. Mobile finds a primary synch signal, the locates the 2nd on the same frequency. It then finds info broadcasted by the BS about the BS. It may get this info from multiple base stations
3. Mobile selects the best BS to associate with
4. Authentication, Set up Data Plane…

Sleep Mode:

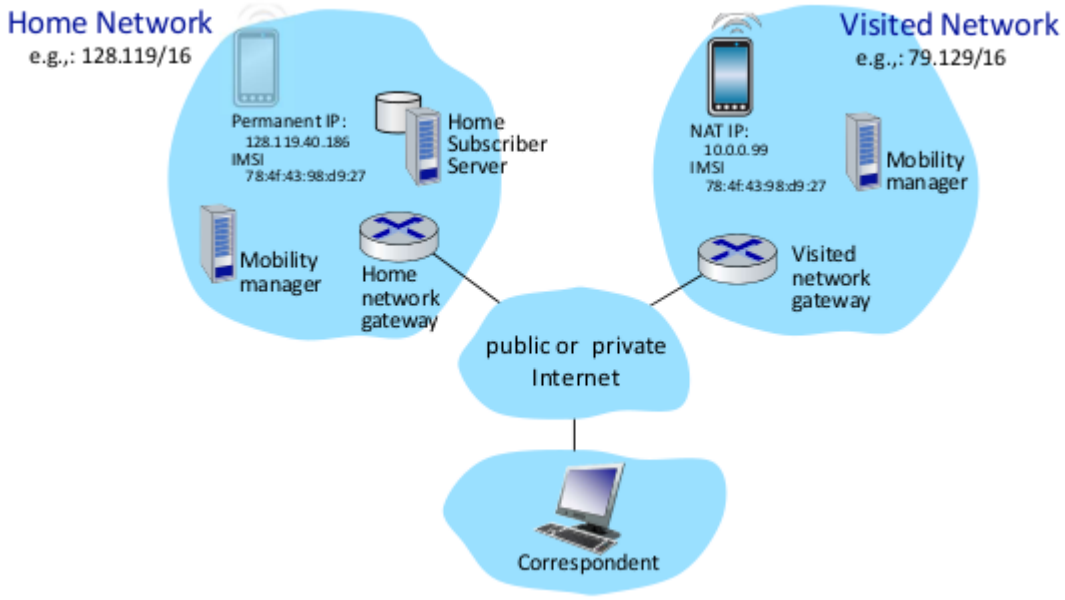
2 possibilities:

1. Light Sleep: after hundreds of mses of inactivity. If you wake up you are interconnected with the cell
2. Deep Sleep: after 5-10 secs of inactivity. I need to re-authenticate to the cell.

Mobility in 4G:
The Home Network is where our HSS is. The Visited network where our UE may move. In the visited network we entered in the roaming. With roaming you allow it to be interconnected with the visited net. For instance, you move to a country from another. In this case you don't reauthenticate, since you don't need to buy another SIM.

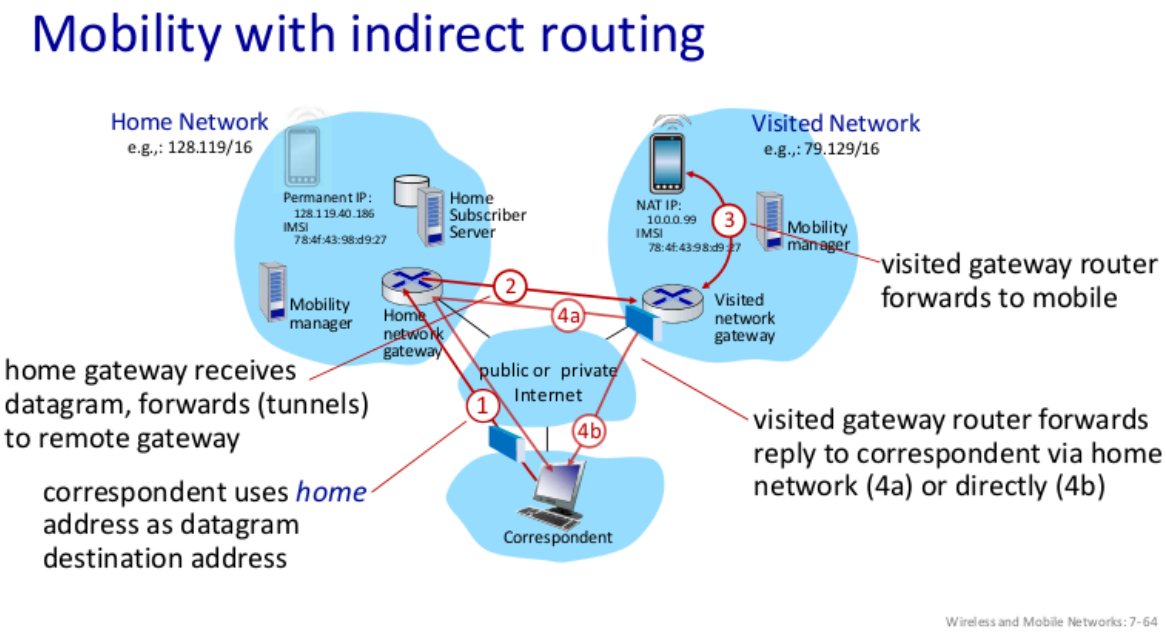


Let's consider this case:



A new user, originally registered in the home network, is temporarily in another country (visited network). Its SIM is fully registered in the home network. In the visited network it needs to have a temporary ID. At the IP level I can't be recognized far away from my home. How can a correspondent in the home network exchange information with the user in the visited network? Two ways:

1. Indirect Routing:



The correspondent sends to the home network. My home is aware of which visited network I am in. So all the IP packets are tunneled from the home to the visited. This is called triangle routing.

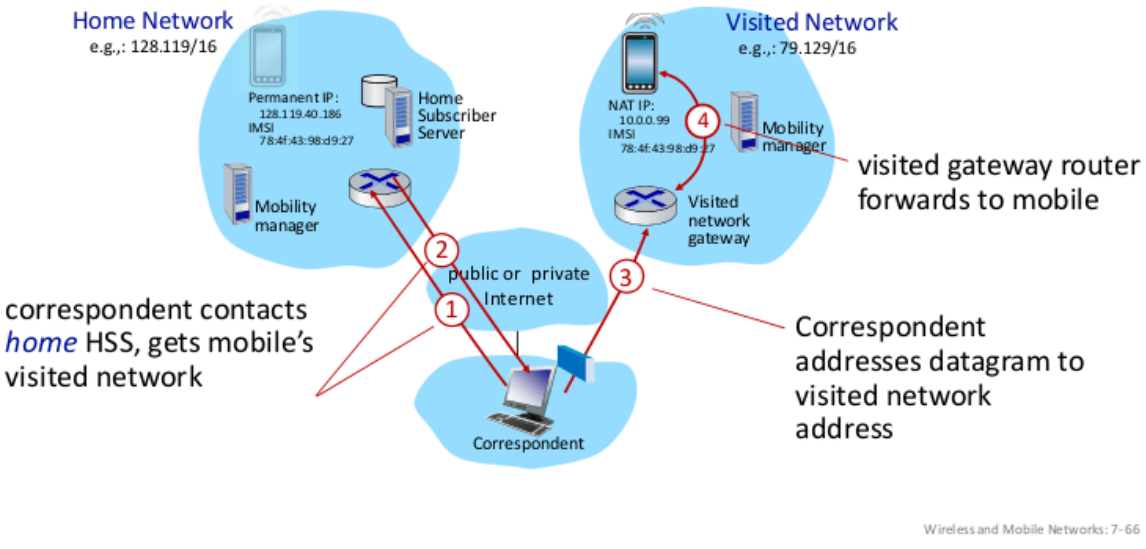
Advantages: there is a chain, if you move to another visited network, the old visited network will inform the home network and the home will know your new position. From the correspondent the position of the user is fully transparent, it just sends information to the home.

Disadvantages: it exchanges more data and at the end there is a higher delay.

2. Direct Routing: the correspondent searches for the final user in the old network, the home knows where the user is and so the home sends back to the correspondent the information of where the user is and the correspondent can communicate with the user directly.

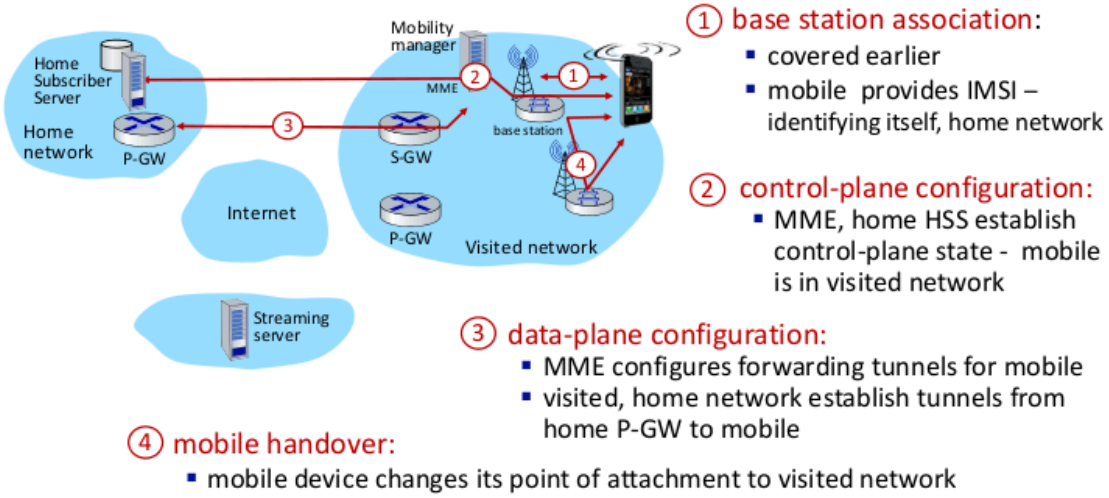
Advantages: not the overhead of the triangle

Disadvantages: the correspondent is aware of the position of the final user (privacy, comunque considera che anche nell'indirect l'operator lo sa quindi non è che hai privacy 100% in quel caso) and overhead in updating each time the position of the user in the correspondent point of view.



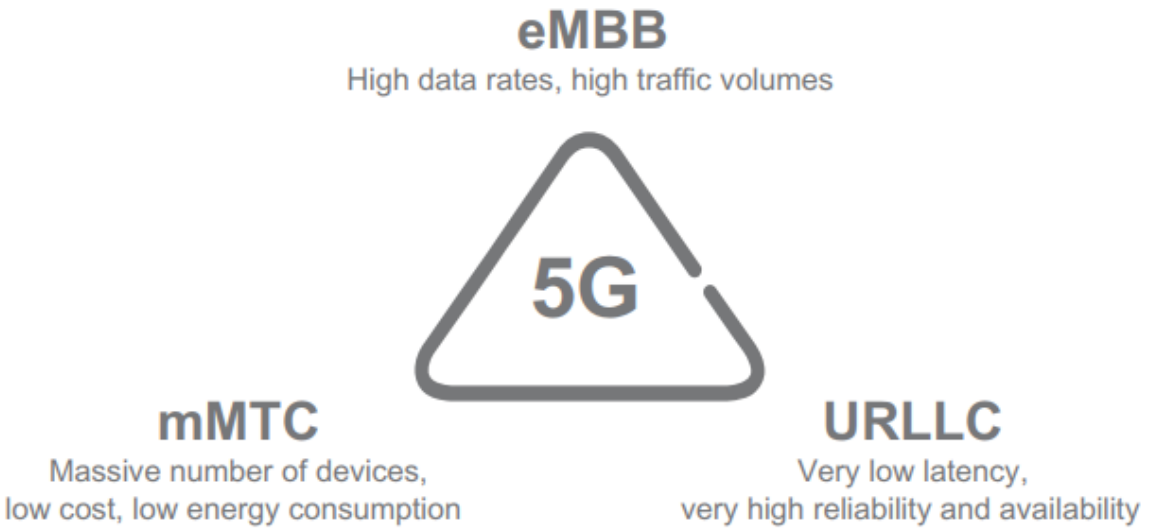
Hand-over in 4G:

The base stations communicate one with the other in order to provide me mobility. This is called hand-over: I need to keep my connection alive when I move.



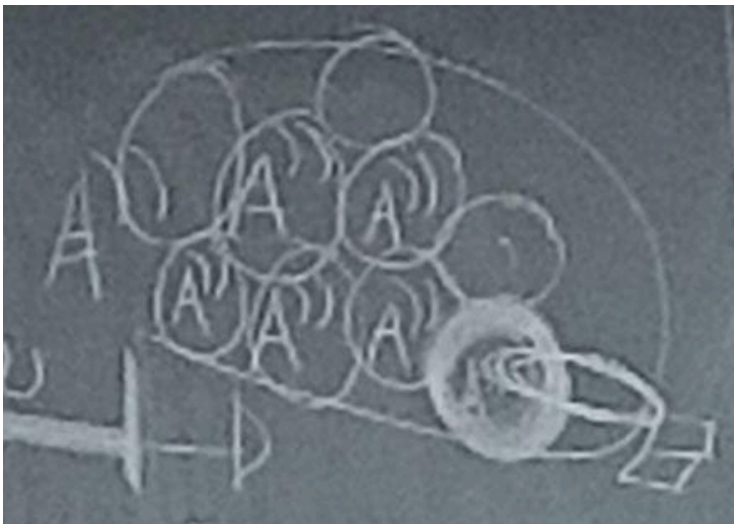
5G:

Application of 5G:



The largest is the bandwidth, the highest is the bitrate. The spectrum is very crowded:

One possibility is to go very high in frequency. In the high part of the spectrum (below 60 GHz but very high) we have mmWave area. In the mmwave obviously the ranges are shorter. So we need a deployment of very dense antenna systems, a lot of them, and they will be very small (picocells). We need the space to put the antennas and the high cost



To increase the capacity in 5G there is also MIMO: instead of spreading the signal all around since it generated interference with the neighbor cell there is the possibility of transmitting to a user with a sort of beam directly intended to the user

MIMO IS THE DEVICE -> beam forming is used

in this way we reduce interference with neighboring cells and the capacity increases

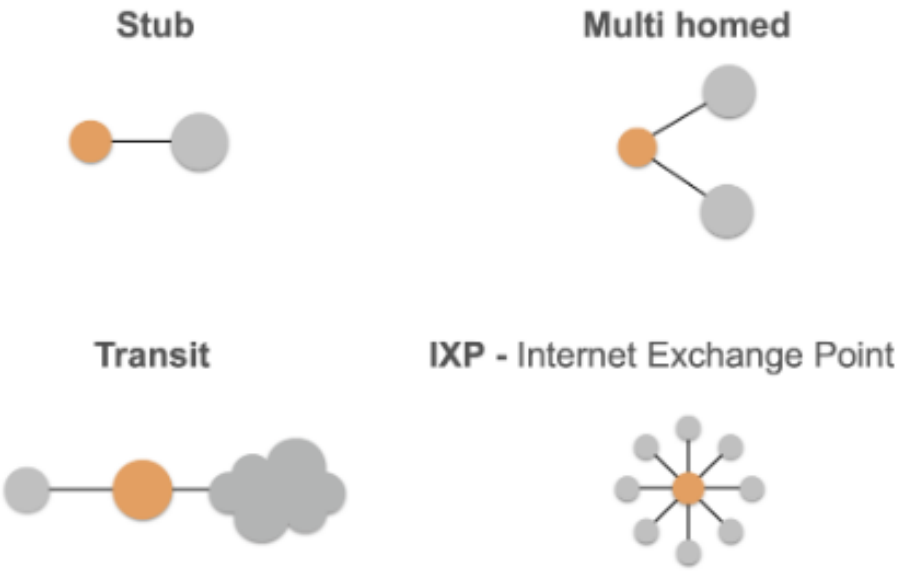
Namex Seminar

18/12

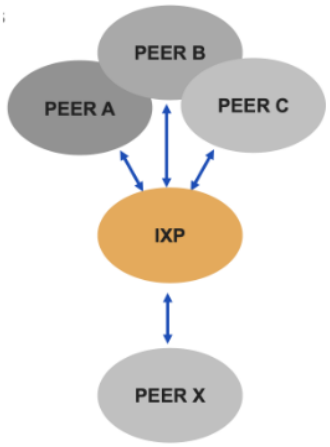
The Internet is a network of networks, basically an interconnection between different participants in the ecosystem. An ISP is one of these participants, examples of ISP are Telecom Italia, Vodafone, Tiscali etc.

Each ISP maintains its AS (Autonomous System). A Regional Internet Registries (RIR) distributes AS numbers and IP prefixes for the ISP. In Europe the RIR is RIPE NCC, in the USA there is ARIN and so on.

Types of AS:

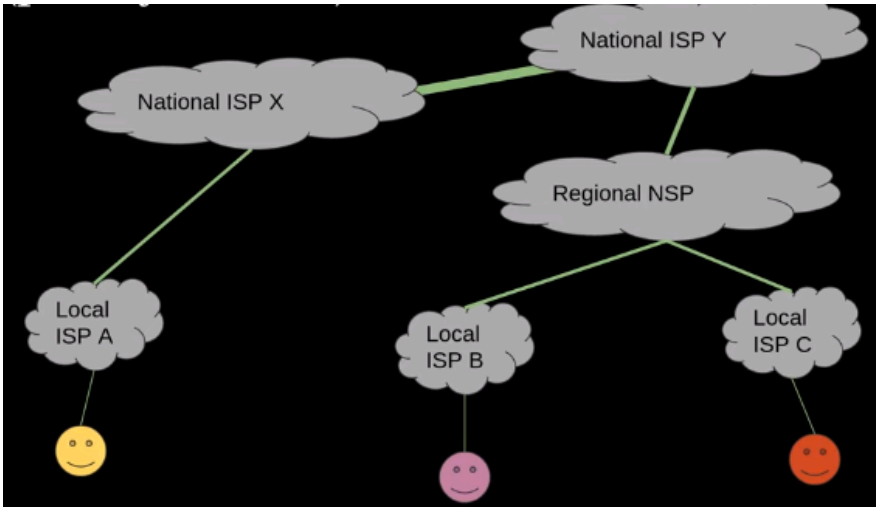


In this seminar we deal with the Internet Exchange Point (IXP). The IXP is a neutral place (Namex is no-profit) where relationships between actors in the Internet ecosystem happen. An IXP is basically a switch, a physical “knot” tying networks together into the Internet.

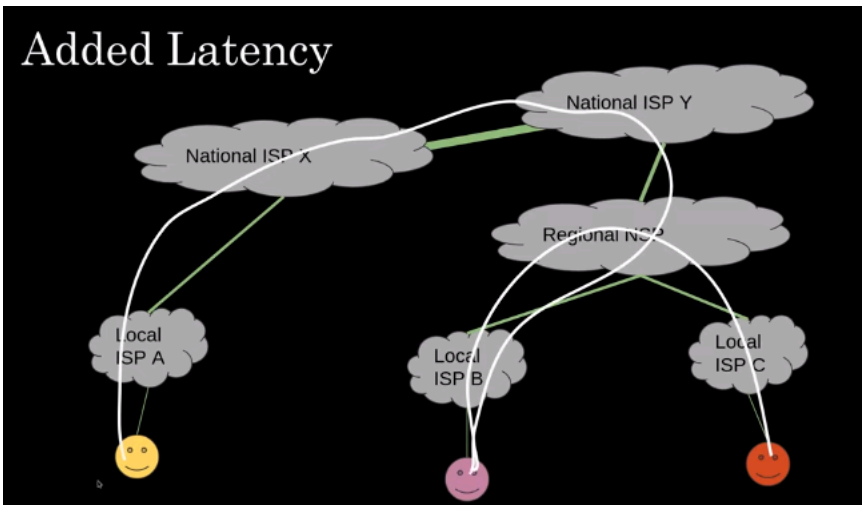


Benefits of using an IXP:

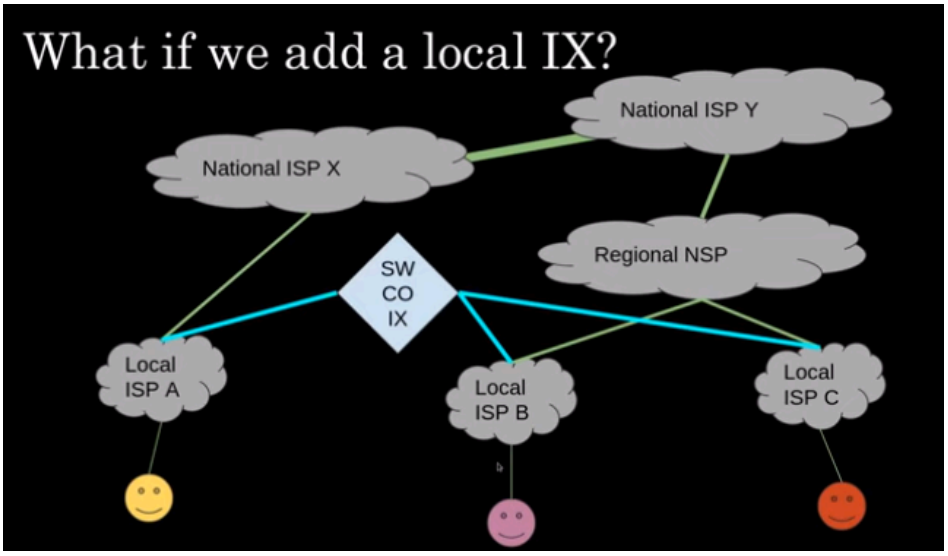
1. Local traffic remains local. Peers can exchange traffic directly, and this is better for the performances (es. latency). Otherwise, there may be a lot of networks you have to go through to reach other users.. For instance:



Let's assume Yellow folk and purple folk are in the same neighborhood, but they have different ISP. Without IXP, the path from yellow to purple may be this



So there is this giro di Peppe even if the two dudes are really close, just because they have chosen different ISP.

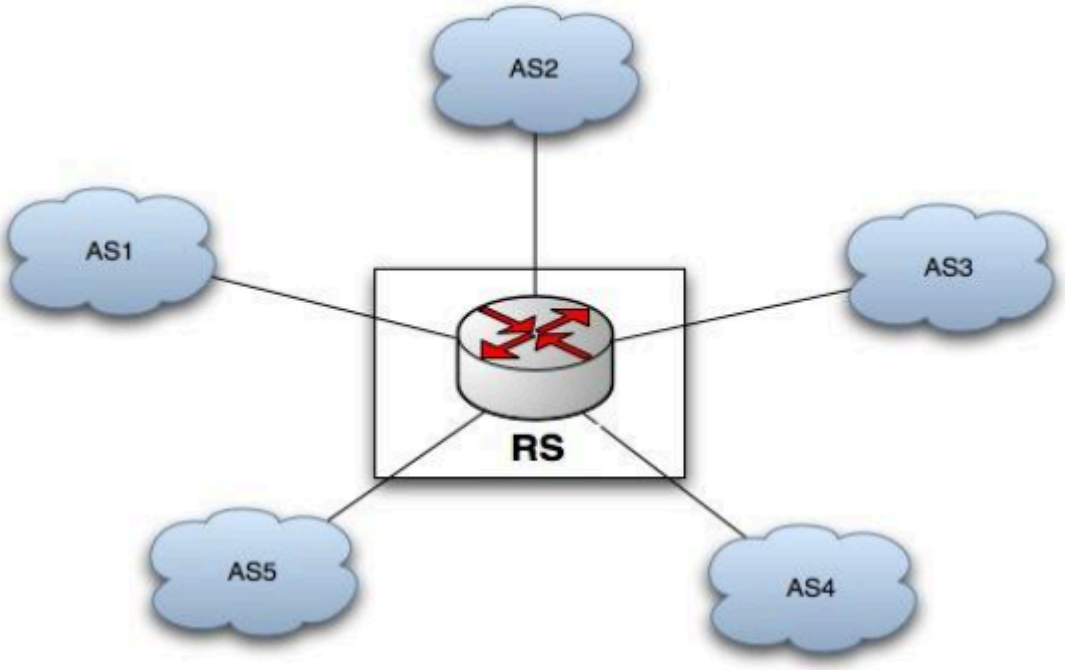


If we put a IXP and ISP A and B become member of it, now local traffic stays local and the yellow user can easily reach the purple without having to pass through National ISPs.

2. Reduces costs for members: without IXP, ISP A and ISP B have to pay someone to get their traffic to the other. With IXP you pay a membership fee[16] but you don't have to pay per bit, or per miles etc.

Route Server:
An RS (Route Server) provides support for the establishment of peering arrangements between IXP peers. A peering session replaces a complex full mesh BGP interconnection, because we just need a single BGP session.

[16] Seen here


AS2
AS1
AS3
RS
AS5
AS4