Encapsulation and Transportation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Layers** |  | **Data Visualization** | **Added Info** |
| 7 | Application |  | Data | Data |
| 6 | Presentation |  |  |  |
| 5 | Session |  |  |  |
| 4 | Traffic |  | Data + L4 Header = Segment | Ports, Method |
| 3 | Network |  | Segment + L3 Header = Paquet | IP Address |
| 2 | Data Link |  | L2 Trailer + Paquet + L2 Header = Frame | MAC Address |
| 1 | Physical |  | Frame | Electrons |

**Switch:** **This is what is really happening:**

|  |
| --- |
| Layer 2: Ethernet II Header  00D0.9752.8936>>0002.17EB.1D01 |
| Layer 1: Port(s): FastEthernet 0/6 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Layer 2: Ethernet II Header  00D0.9752.8936>>0002.17EB.1D01 | | | | | | |
| Layer 1: Port(s): FastEthernet 0/6 | | | | | | |
|  | Layer3 Info-Paquet | | | | |  |
|  | Layer4 Info-Segment | | |  |
|  | Layer5 Info |  |
| Layer6 Info |
| Layer7 Info-Data |
|  | | | | |

**Difference between TCP/IP and OSI:**

They are basically the same thing, because the TCP/IP protocol does the exact same thing but compresses everything that we do in the presentation and session layers inside the Application Layer, or layer seven.

**Presentation Layer:**

As its name indicates, the presentation layer makes the data presentable in a format, in different types of data. One way of thinking about **data** **format** is file format, like “.jpg” or “.txt” for example.

The presentation layer is also responsible for the **encryption** of the data received. This makes sure that a hacker can’t see what the data is that’s being sent. One common encryption method is **SSL** or secure socket Layer

**Session Layer:**

This is a very important step when we are stablishing communication with a web page for example. This layer oversees creating a communication channel between the two devices so they can share some determined information that is vital for this, just once per session. Once you close your web browser and open it again or restart your device this session will be closed and it will be created next time you need to create a channel between you and a server, like **YouTube**.

The session Layer keeps communication ongoing and when it starts the channel uses something called the **three-way handshake**.

**Three-Way Handshake:**

This is how the channel is established, the session is created. First the one who is sending the data sends a **SYN**chronization message, then the one who receives the data sends a

**SYN**chronization **ACK**nowledgement message, and then he responds back with an **ACK**nowledgement message.

>>>**SYN**

<<<**SYN ACK**

>>>**ACK**

So, what happens next?

Now bellow the Session Layer we have the transportation layer. Here we need to choose one of the transportation method TCP or UDP. But how do TCP and UDP really work?

TCP and UDP uses :

The thing here is that **TCP** ( Transition Control Protocol ) sends a message and listens for an answer from you, and if you don’t reply to that message YouTube will send that information for you again. UDP in the other hand, it doesn’t care if you received that data or not, it would continue to send you more and more data without knowing if, for example your internet at a very little moment was kind of fuzzy and you didn’t get that packet.

***When is better to use TCP or UDP?***

TCP is used for information that are vital, and no real time, like an HTML file for example, or when we download a file from a server. UDP on the other hand is used for things like videogames, YouTube videos or Facebook Live, Instagram Live and those kinds of things.

To give a better look to it, lets see how it happens using **Wireshark**:

Graphical user interface, text

Description automatically generated

Here we can clearly see how 173.194.191.167 (YouTube), is establishing a Three-Way Handshake with our device by using the, SYN, SYN ACK and ACK messages. As we can see if we want to do a three-way handshake, we need to use the TCP protocol, as we can see in the third column.

Ports:

Ports, in our devices, are like the “ports” in an airport, we can have multiple airplanes going and coming to and from different parts, at the same time. Just like in that case, our devices have ports that are used for obtaining and sending different types of data and doing so all at once.

We choose from Port to Port to use the right one by following one criterion: What Protocol Are we Using?

There are lots of protocols for lots of way of accessing a server, like HTTP, if we want to access a web page. This happens to be located in port 80. If we want to use HTTPS, then we’d use port 443. But we don’t only want to get some information, sometimes we want to post something, like we do in Django, or sometimes, we’d also want to SSH, yes SSH is another protocol and is located in port 22.

Now let’s see something cool. When you access a website, that little thing you type at first, is the port, and the url, is going to be solved by your DNS server to find the IP address corresponding to that specific address, like in <http://www.google.com>. HTTP means, go on port 80, if we instead do https, that will mean go there, but use port 433.

If we go to google and google it, we will have all this info available.