01-04 Generalities ArchitectureV1 (0:01) Okay, let's continue with generalities, and now we're going to see something very important in (0:07) networks, which is the architecture. Architecture exists or appears due to the complexity of all (0:16) the activities that need to be done within a network. So, what researchers (0:22) around the topic have been developing, let's say they've been evolving, is how to address (0:30) all the aspects related to the network in an organized and structured way (0:35) that allows us to easily tackle specific issues without affecting the rest of the network's organization (0:41) and making the problem a bit less complex to solve different aspects. (0:48) So, let's look at some topics around this. Architecture, the first thing we need to do (0:54) is define what network architecture is. Architecture is a set (0:58) of layers, protocols, and interfaces that facilitate the design, programming, operation (1:09), and maintenance of the tasks managed by a network. That is very important. What network architecture aims for (1:17) is to make it easier to design (1:23) networks, make changes to the design, do the programming that is behind all (1:28) the design we define, make operations easy to manage, and make network maintenance relatively simpler than if everything were (1:39) consolidated. Network architecture includes design principles, physical configurations, (1:45) functional organization, operational procedures, data formats required for (1:54) design, construction, modification, and operation of a communication network. (1:59) So, from these three elements—layers, protocols, and (2:05) interfaces—we manage to configure and understand how a network works properly. What is a set (2:13) of layers? Layers define how the network functions are structured. (2:22) The layers help manage complexity because what I do with the layers is decide (2:28) how to divide the network's functions. So, if the network has to do 50 things, then (2:34) they tell one layer, you handle five of the things the layer has to do and (2:39) another layer handles these ten, and another handles these eight, reducing the (2:46) complexity in each layer. An example of this is, for instance, when you organize a party. (2:54) When you organize a big party, if you take care of all the aspects (2:59) related to the party, imagine a big event; if you handle the entire organization (3:05), it becomes very difficult to keep everything in mind. You have to send (3:10) invitations, someone has to worry about the drinks, other people perhaps about (3:16) the food, someone needs to be checking the music, the desserts, the venue, the decorations. (3:25) There are many things to worry about. So, if I have one single (3:31) monolith handling the food, checking the desserts, the organization, the arrangements, (3:40) the music, everything becomes more complicated. On the other hand, if I divide the functions and say, (3:45) someone takes care of the food, someone handles the drinks, someone manages the desserts, (3:51) another person takes care of the music, I can divide those functions, and the person handling (3:56) the food will focus on buying the ingredients, determining how many dishes (4:01) to serve, what kind of dishes to offer, and when to serve them. One can organize (4:07) this more simply. The same applies to network layers, which is why it is said (4:13) to reduce complexity. Each layer assigns functions to the upper layer so that they can be (4:23) used by one another. We will see this in a diagram. Each layer has to perform the (4:31) defined procedures, which I mentioned before, and it allows for modifying the functions (4:34) or protocols within a layer as long as it doesn't affect another layer. So, that has to (4:42) do as well, and it's one of the key roles of layers: if something changes inside (4:50) the layer, the other layers shouldn't be affected, and that is very important. It's similar to what I mentioned with the party; if they decide to serve chicken instead of beef, the person handling the music shouldn't be worried because it doesn't affect what they're deciding. (5:07) In general, that's the goal of layer design: to ensure that changing one layer doesn't affect the operation of another. That's what it seeks and allows for in networking; it doesn't matter if some things change in life, (5:20) the rest of the network operates well, maintaining the operation of the other layers. The other element is protocols, which define the rules and conventions followed in communication between layers of different machines, covering syntax, semantics, and timing. (5:42) What a protocol aims for is that if a computer has an X layer operating in a certain way, the identical X layer in another computer will understand what that originating layer is communicating. (5:58) Using the party example I gave earlier, if I assign someone to handle the food, they will likely communicate with others who understand food, and when they say they need a pound of something, those people will understand the request. (6:17) Different from the music manager, who would need to discuss other subjects to be understood, or the person handling invitations, who might discuss paper types, fonts, colors, or packaging with others who have similar roles, that will aid in their operation. (6:39) In the next diagram, we will better understand how this works within the network. Finally, interfaces describe how communication occurs between layers of the same computer to exchange information between adjacent layers, establishing a boundary between two systems or programs where signals and instructions (7:09) are precisely defined. In networks, interfaces tell the upper layer how to use my functions, like making a public function, method, or program accessible so others know what parameters (7:37) they need to understand and what functions to use for interacting with a particular layer. In this illustration, we discuss protocols further, which, as mentioned, are rules and conventions for communication. For instance, when humans converse, typically we greet first, then establish communication; in the case of computers, a similar process is needed for them to interact. (8:07) This is what protocols do. Here is a more formal definition: a protocol defines the format and order of messages, specifying how a message is structured, (8:21) that it must start with a name, followed by a greeting, then the message itself, and end with a farewell, outlining the exact protocol we need to follow. It's like the rules of the game. And the order of the messages, determining which comes first, which comes later, between two or more entities communicating, and the actions to take during transmission and reception of messages or events. (8:59) This is essentially how network architecture is viewed, being a layered structure. Each layer—layer 1, layer 2, layer 3, layer 4, and layer 5—has its functions. (9:18) If I change something in layer 2, for instance, layer 1 shouldn't be affected, nor should layer 3, or any of the other layers. Any changes within that layer shouldn't impact the others. (9:34) Protocols define how identical layers communicate between two different systems. Here, you have Host 1 and Host 2, and their communication relies on protocols. (9:48) Interfaces define how two adjacent layers communicate. For example, layer 4 offers interfaces to layer 5 so that layer 5 can communicate and send information to layer 4, and so on. (10:12) Similarly, layer 1, for instance, provides a set of interfaces to layer 2 for data exchange between them. And finally, there's the physical medium that enables interconnection in some way. (10:30) Many network architectures have been created. Various research groups or people who have worked with networks in the past have devised different approaches to network architecture. (10:52) Some have become actual architectures, while others remain models. The distinction lies in that a model is proposed by a group of people, an organization, or a research group but remains just a model and isn't implemented. (11:12) An architecture, on the other hand, is a model that has been implemented. So, the architecture is in place, while the model remains a recommendation on how things should be done. (11:29) One well-known model worldwide is the OSI model. The OSI model is defined by

ISO. This model outlines seven layers. In the previous slide, I showed five layers. The OSI model defines seven layers. (11:52) These seven layers are widely accepted as a good way to distribute network functions and are shown on the left as a standard way to review network architectures developed by companies. (12:13) For example, IBM came up with this mechanism. Here's the one used by the internet. And, well, around this, you see some other steps. (12:30) What I want you to understand is that many ways have been devised to organize networks. (12:38) (12:38) The TCP/IP architecture is the one that ultimately prevailed, and the reason it did is because the internet is used. And so, they have defined all these protocols that are seen here, somewhat, that are related to the network topic and how the network's functions should be distributed. (13:05) Generally speaking, almost all architectures have assumed that the first layer, at the bottom, and the second layer are the same, in a manner of speaking. (13:21) I mean that they, and for example, the MPE model, that is true, and, well, in general, it happens similarly. They focused on working from a point upward and said, whatever the rest of the people are doing in research down here, I trust it. (13:38) Keep doing it, and I'll build upon that. So, almost always, network models or architectures assume that the lower part of that model is somewhat standard, and they propose things for the upper part. The IBM model also did something similar. (13:55) It is very normal for this to happen with all models, as you can see there. Or architectures, because in this case, the IBM one was true, the TCP/IP one, which is the Internet's, is also true. Here we have Poltoca, which is also true. (14:12) Well, all those others are valid, they are architectures that at some point were believed to be used, that should prevail, but that in the end, the internet gradually surpassed and left behind. (14:28) Model 12, as I mentioned, is a seven-layer model. Physical, link, network, transport, session, presentation, and application. And as I mentioned, it is a model because it was not implemented, it was simply a proposal by ISO. (14:48) This was defined by ISO, which said how the functions of a network should be organized and placed them here. So, they say, the physical should be these things. Link should be these others. Network should be these others. And so on, they made a proposal. (15:08) Some setups, some architectures resemble this. It was a time, around the '50s, when all researchers wanted to organize the network in a simple way, and many proposals came out, and in some cases, they are very similar because they all asked what the network should be, what the network's functions were, and how we organize them. So, they are somewhat similar to each other. (15:37) The TCP/IP architecture is the one defined when the ARPANET model and the whole Internet theme were invented. It was defined by the universities that worked with the United States Department of Defense and designed a four-layer architecture, which can also be seen as five, where there is something they called network access. (16:07) Then the layer of what they call Internet, transport, and application. This network access, as I mentioned, are the lower layers of the model, and it still refers to the physical layer, the link layer, and then here, Internet, which is mentioned as network, and here, transport and application. So, one can see a model of five layers or four layers. (16:34) Purely speaking, if one thinks purely, the Internet model, the TCP/IP architecture, is really four layers, but many and some modifications that have been made speak of five layers, and we will stick with the five-layer model. (16:53) Here we see a comparison between the OSI model, which is the most general model, and the TCP/IP architecture. In this one, we see the seven layers; here, we see the five layers. We see how the physical layer is more or less the same, here it's a little different. (17:16) What is called here Internet is very much what is seen here. The transport layer of the TCP/IP architecture takes everything that transport was defined as in the OSI model and part of what is session, and the application layer of the TCP/IP architecture takes part of session, plus presentation, plus the application layer of the OSI model. (17:40) If one thinks from the perspective of how implementation is done, some parts are implemented in hardware, other parts in hardware that can be programmable, which is the firmware, and another part in software. (17:58) And so, in hardware, we would stay at a maximum, reaching up to layer 3, that is, up to the network or Internet layer. In firmware, we go from the link layer to the transport layer, basically we can go. And in software, we can also go from the link layer, of course, to the application layer. (18:20) And the implementation, whether physical, firmware, or software, in this third part, is a matter of whose responsibility it is to be concerned about it. (18:36) So initially, what was said is that it was the responsibility of the operating system to be in charge up to here, meaning taking care of the physical layer, the link layer, the network layer, and some of the transport layer. (18:48) And in the user portion, we should start from the transport layer, a bit, seeing it from the TCP/IP architecture perspective and the application layer. (19:04) Now, this topic here that should be handled by the user is really not the end user, many times it is the applications used by the end user, and the end user builds on top. (19:19) It is more or less the structure, and what has also happened sometimes is that the operating system has been, I don't know, gaining ground a bit and then starts covering more than it was initially defined to, moving upwards. (19:33) To make the network more transparent for the user and to provide services almost without thinking too much. (19:44) Here is an example of an artistic architecture in a network, and this is also important. (19:50) Generally, in end devices, in hosts that can be a cell phone, a computer, or a television, we speak of a model, as we mentioned, of five layers. (20:04) That is normal, but in the part of what is the internet, the network portion, known as a subnet, which includes all these elements inside that allow the interconnection of hosts. (20:18) All these elements that are included here and have an organization only have three layers. (20:24) So each node in the network, in the subnet, only has three layers, it does not have five layers. (20:34) And we will see why it only needs three and not five. (20:38) Because it is a matter of functional need within the network. (20:43) So intermediate devices in the network can have three layers, and there are even devices that have only two layers or others that have just one layer. (20:54) We will see throughout the course which have how many layers depending on the functions they fulfill within the network. (21:03) Here we have arrows of interconnection between layers, which are the interfaces, but we also have arrows of interconnection between layers with another device, and these are the protocols that we will mention here. (21:22) It is also important to talk about the data units in those layers. (21:26) And there we have to say that each layer, when we are trying to transmit data over the network, must have control information to account for the functions it needs to fulfill. (21:45) So data units are known as PDUs, which is Protocol Data Unit. (21:55) And then the PDUs of each layer have a different name. (22:01) In the application layer, PDUs are called messages; in the transport layer, they are called segments; in the network layer, they are called packets or datagrams; in the link layer, they are called frames. (22:14) And at the physical level, one speaks of a bitstream. (22:17) So when there is data that the user wants to send, it is handed to the application layer, the application layer adds a header plus the data the user sent, and that is called a message. (22:36) All of this, ready to be sent, is handed to the transport layer, and now a header is added, and it is called a segment, and so on. (22:47) Here they are called packets, and a header is added. (22:51) And in the link layer, data is returned, a header is added, and data is added at the end to mark the start and end of the official data, and here comes the bitstream. (23:09) It is like when one sends a package from one place to another, for example, if I am sending a glass cup, first I wrap it here, then I would wrap it in paper, then I add bubble wrap, then I put it in a box, then I wrap it in another box, here I add the box with the address, and then I send it. (23:32) When it reaches the other side, they will have to unpack each of those layers, gradually unpacking the user's message until it delivers only the data itself to the user, but each layer has a specific name. (23:54) And here I show you a bit of the data unit operation, how they start forming from here to the user's message, and they are handed to each layer, each adding headers, and when they are sent to each element, notice that this element has only two layers, this one has three layers, so the data will need to be packed and unpacked until they finally reach their destination. Ready? Well, there we already saw a little about what the architecture is like and what the data units that are the CPDUs are like.

Now we are going to look further at what the functions of each of these layers are to understand how each one works and how we are going to work the rest of the year. Thank you so much.