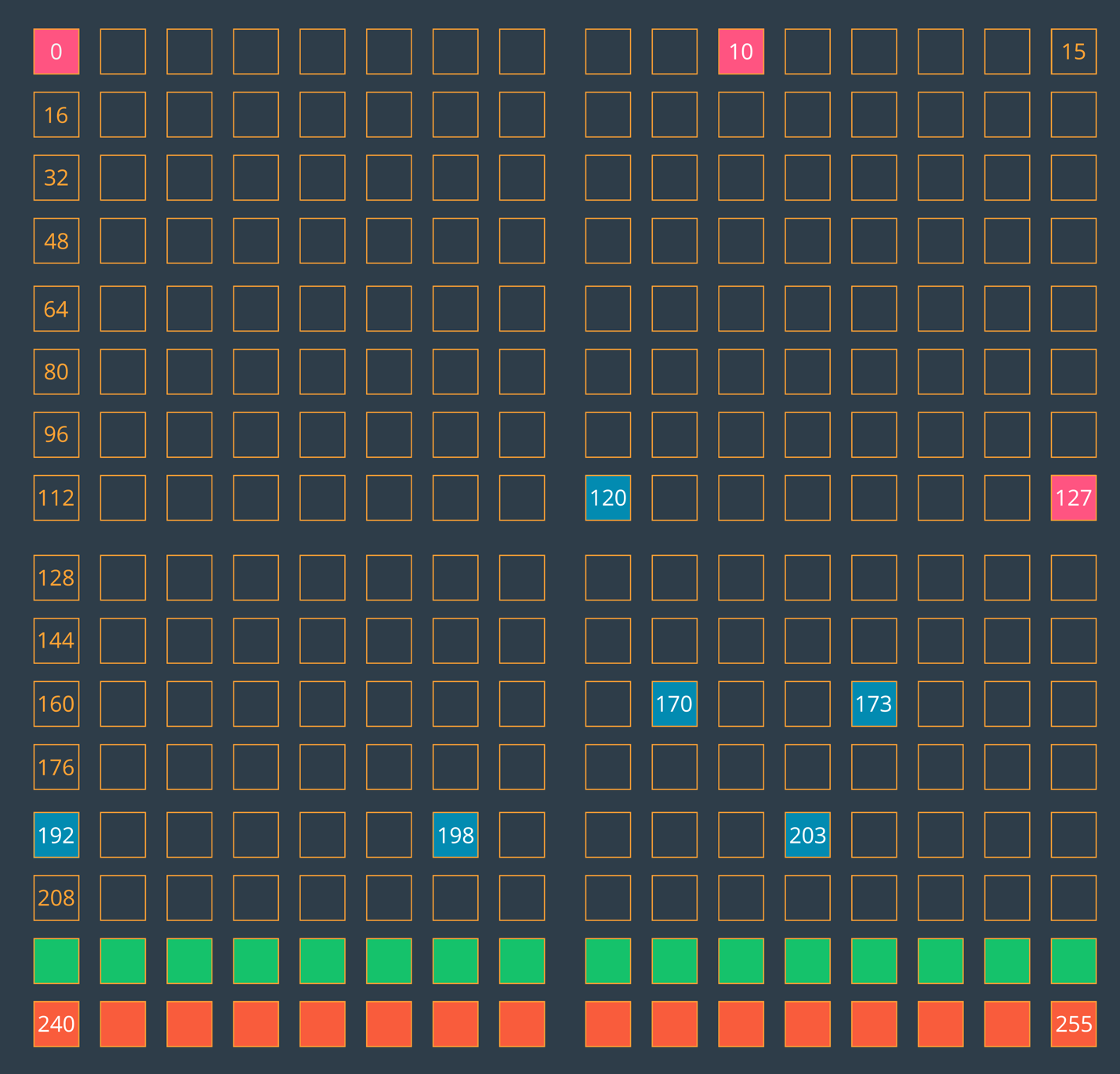
**Addressing Space**

Here's a visualization of the entire IPv4 address space. Each square represents one possible value of the *first octet* of an address. For instance, the square on the top right, labeled 15, represents all the IPv4 addresses that start with 15 as their first octet (e.g. 15.72.9.277).



Visualization of the IPv4 address space. Each square represents the first octet of an IPv4 network address.

* The pink squares (0, 10, and 127) are blocks that are entirely reserved.
* The blue squares are blocks that are *partially* reserved. For instance, not all of the 192 block is reserved, but some of it is.
* The entire green row (starting at 224) is set aside for [IP multicast](https://en.wikipedia.org/wiki/IP_multicast" \t "_blank).
* And the entire orange bottom row (starting at 240) was [originally set aside for "future use" but was effectively lost due to being blocked as invalid.](http://packetlife.net/blog/2010/oct/14/ipv4-exhaustion-what-about-class-e-addresses/" \t "_blank)

**Review of Binary Conversion and the AND operator**

If you watched that video and wondered, "How do I convert to binary?" and "What's an AND operation?", don't worry because the following resources will explain the basics to you.

If you'd like to learn how to convert decimal to binary by hand, read [this tutorial on the subject](https://www.electronics-tutorials.ws/binary/bin_2.html" \t "_blank).

Once you understand how the math works and have practiced a conversion or two, you'll probably want to use an online conversion tool for efficiency's sake. The following two tools were used to perform the conversions seen in the video,

* [IP Address to Binary](https://www.browserling.com/tools/ip-to-bin)
* [Binary to IP Address](https://www.browserling.com/tools/bin-to-ip)

Additionally, the process of determining the network address given an address of a host on the network and the subnet mask requires you to perform a logical AND operation. The AND operator compares two inputs to produce an output. In the case of the address and mask, it compared each bit position individually.

The following truth table summarizes which inputs produce what output.

| **Input 1** | **Input 2** | **Output** |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

You read the table above from left to right. In the first row, Input 1 is 0, Input 2 is also a 0. If you do a logical AND of 0 and 0, you'll get another 0. The second and third rows are both doing a logical and of a zero and a one (albeit, in different orders) and both result in 0. The last row is the interesting one; a logical AND of 1 and 1 equals 1! So the logical AND is actually very simple - the output is only 1 when *both* inputs have values of 1.

Happy Converting!

**Reserved Addresses on a Netblock**

If you recall, the number of addresses that are available for use in a netblock were reduced by three, because those three addresses are reserved for something. Well, what are they?

* The first address (.0) is used for identification of the network,
* the follow address (.1) is often assigned to the router,
* the last address (.255) is called the broadcast address. Anything sent to the broadcast address will be sent out to all devices on the network.

**Additional Resources**

Some basic reading on subnetworks and Classless Inter-Domain Routing (CIDR):

* [Subnetworks](https://en.wikipedia.org/wiki/Subnetwork),
* A good read on [CIDR](https://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing" \t "_blank) with a helpful table on all IPv4 CIDR blocks.

And several technical reports on the following subjects:

* [RFC IPv4](https://tools.ietf.org/html/rfc1878),
* [RFC IPv6](https://tools.ietf.org/html/rfc5942),
* [RFC CIDR](https://www.rfc-editor.org/rfc/rfc1519.txt).

# OSI Model



Two different parties trying to communicate across a "network".

In the image above, Richard and Julia are trying to communicate over a tin can phone. Earlier, you learned that there are three main components in a network: the medium, the addressing, and the content. In this tin can phone example, the medium is the cord connecting the phones together. We're already directly connected to each other so we don't have to worry about the addressing, and the content is what we're trying to tell each other.

Let's take a moment to think about how we communicate the content to each other. For example, how does Richard know when Julia is finished talking so that he can reply? It would make communication a lot easier if they both agreed on a set of procedures to do each time they start and stop talking. For example, before sending her main message, Julia could begin the communication channel by saying "Message is starting". Then, so that Richard knows she's finished communicating her message, Julia, when finished sending the content of her message, could follow up with "Message is finished".

This is a simple example, but the important thing here is that these set of procedures are **protocols**, and they can be used to help standardize communication across a network. In this example of the tin can phone, it's two people using soup cans to send a message across a cord. But what if you swap out the people with two computers and swap the cord out with a network cable? There are still two different parties that are trying to send a message across a network. How should the two computers communicate effectively and efficiently? What sort of protocols should they use to make sure their messages are sent and received correctly? There can be many, many more devices than just these two computers. In this internet-connected world, virtually anything can be a network device!

So with the flourishing ecosystem of vendors (e.g. different manufacturers, different devices, etc.), we need to ensure the interoperability of both devices and the medium across which these devices communicate so that network communication can function effectively.

How can we ensure that any device can send any kind of message across any kind of medium to some other device? Well, we need a standard that all parties can agree upon so that communication across a network can occur. In fact, as you might expect, we'd need a lot of standards for this to occur. There are standards for cables, their connectors, how data is packaged, how one device knows when to send to another device, how data is converted to electrical signals, how messages are broken up into chunks, how two devices are able to send an error-free message, and so on. There are many standards for all of the different parts of network communication.

And this brings us to the **OSI model**! OSI stands for:

*Open Systems Interconnection*

The OSI model is a breakdown of the various aspects of network communication. It is concerned with the ability of different systems to be connected with each other. It's a conceptual breakdown of all the parts that are involved and groups them into logical sections. Let's check them out.

## Layers in the OSI Model

We're going to look at all of the layers in the OSI model. But before we do that, I just want to point out again that this is a conceptual model for a network system to handle communication; it's not a concrete example of how network communication happens. The OSI layers will be listed here for you to look them over, and then there will be a video to walk through them in more detail. You don't need to memorize this information. Instead, you should focus on absorbing how the layers are arranged. Later, you'll study these layers in more detail.

There are 7 layers in the OSI model. We'll start at the top-most layer, layer 7, and work our way down to layer 1.

Layer 7 - Application

Layer 6 - Presentation

Layer 5 - Session

Layer 4 - Transport

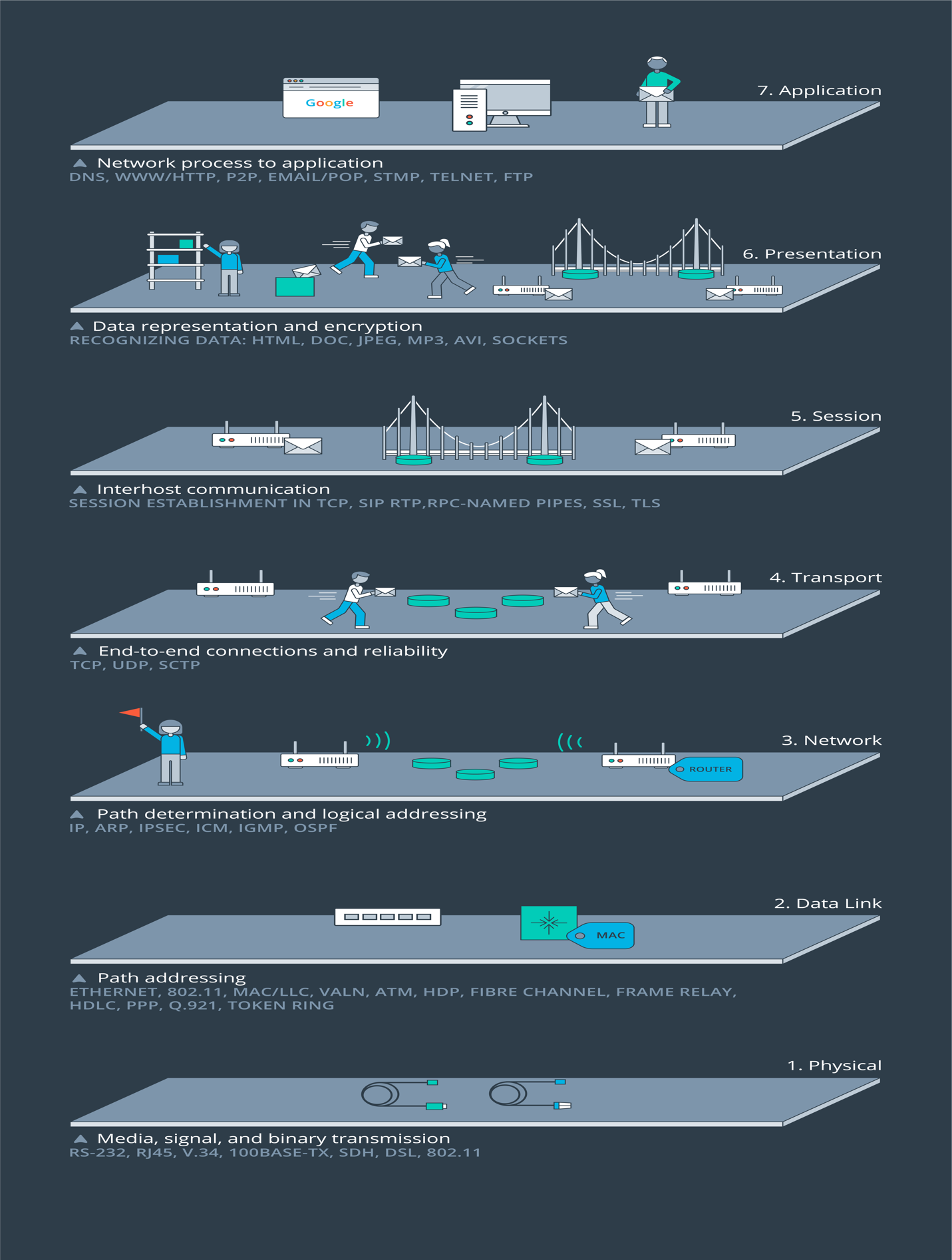
Layer 3 - Network

Layer 2 - Data Link

Layer 1 - Physical

Remember that the OSI model is a conceptual model of how network communication occurs. For network communication to happen, the layers must work together to transmit a message from one device to another.

## Standards & Protocols



The seven layers of the OSI Model.

As we've been talking about, the OSI model is a conceptual model and helps describe data in many ways. It provides a standard for how network communication occurs, how data is transferred, and provides a way to group similar protocols together.

For example, network devices can be grouped at several different layers in the OSI model. Layer 1 of the OSI model deals with the physical and electrical specification for devices. Items such as physical cables, wireless signless, the network identification card (NIC) inside a computer are all grouped at the Layer 1 level. But even though Layer 1 is called the physical layer, that doesn't mean that every physical device shows up there. For example, a switch is a Layer 2 device and a router is a Layer 3 device.

The OSI Model also helps group network protocols together. Here are some of the most common protocols at each layer:

* Layer 7 = HTTP
* Layer 6 = SSL and TLS
* Layer 5 = socket
* Layer 4 = TCP and UDP
* Layer 3 = IP and ICMP
* Layer 2 = ARP
* Layer 1 = 1000BASE-T, DSL, DOCSIS, 802.11a/b/g/n

The information that goes along with the OSI Model is vast and deep, and we're not going to go into more detail on the standards and protocols than we have right here. I just want you to be aware of another aspect of the OSI model and to see from another perspective how it is a conceptual model to help group information.

## Sibling Dependence

In the OSI model, a layer is only concerned with the layers above and below it. For example, layer 3 (the Network layer) is only concerned with the two layers that are on either side of it:

* Layer 4 - the Transport layer
* Layer 2 - the Data Link layer

Layer 3 will receive data from Layer 4 and send it to Layer 2 or it will receive data from Layer 2 and send it to Layer 4.

Layer 7 (the Application layer) is at the very top of the OSI model. Since it is at the top, it is only concerned with sending data to Layer 6 (the Presentation layer) or receiving data from Layer 6.

A single layer does not need to know the whole path that data takes through the OSI model. It just need to know about the layers above and below it that it needs to interact with. Now, why is this important? Each layer in the OSI model is an [abstraction layer](https://en.wikipedia.org/wiki/Abstraction_layer" \t "_blank). An abstraction layer is:

*a way of hiding the working details of a subsystem*

For example, the way that the data is formatted, processed, and changed in Layer 2 does not matter to Layer 4, because Layer 3 abstracts those implementation details away. All Layer 4 has to worry about is how it needs to receive data from Layer 3 - it doesn't have to worry about the specifics of Layer 2 (since that's Layer 3's job).

Earlier, I said:

*Layer 3 will receive data from Layer 4 and send it to Layer 2 or it will receive data from Layer 2 and send it to Layer 4.*

When does data move down the layers from Layer 4 to Layer 3 to Layer 2 and when does the data move up the layers from Layer 2 to Layer 3 to Layer 4? These two processes are called **encapsulation** and **decapsulation**. Let's take a look at them in more detail.

## Encapsulation and Decapsulation

Remember that the OSI model is a conceptual model of how data is packaged and sent across a network to another device. In this scenario, let's assume that there are only two devices that are trying to communicate with each other. The OSI model is used on both devices.

As the data is being packaged up on a sending device, the data moves down the OSI model; this process of the data moving down the OSI model is called **encapsulation**.

When the data is being unpacked on a receiving device, the data moves \_up\_ OSI model; this process of the data moving up the OSI model is called **decapsulation**.

Let's see each of these in more detail. First, we'll start with encapsulation.

To recap, when data travels down the OSI model, the data from a higher layer becomes the "payload" for a lower layer. When the data moves from a higher layer to a lower layer, the data is wrapped in new information called a "header" and a "footer". This is where it gets the name "encapsulation"; because the data is wrapped (or encapsulated) with new header/footer layers.

Conversely, the opposite happens when traveling up the OSI model. As the data moves up the OSI model, these header/footer layers are removed. This is where it gets the name "decapsulation".

## Network Transmission

Hopefully you're seeing how the OSI model can helps us to understand how network communication happens across a network. You saw how encapsulation works on the sending device. Then you saw how decapsulation works on the destination device. We looked at both of these in isolation so that we could zoom in on exactly what's going on with encapsulation and decapsulation. But now, let's zoom back out a bit and see how data moves from one device to another in a network.

We're going to look at one device sending data from one network to another device in a different network. We'll be using all of the information that we've covered up to this point, so this video might be a bit complicated. You might need to rewatch the video once or twice to let these concepts sink in.

To recap, as data moves across a network, each devices looks at specific header information to determine if the data belongs to itself or if it should forward the data on to another device. One thing to note here is that this process of moving up and down the OSI model (the process of encapsulation and decapsulation) happens over and over as the data moves from one device to another. To clarify, though, the information doesn't move all the way up through all of the OSI layers at each device, it only does that at the final (target) device.

## Summary

In this section we looked at the layers of the OSI model, how data moves down the OSI model on a device, how it moves across a network, and how data moves \_up\_ the OSI model on the target device.

The OSI Model layers are:

* Layer 7 - Application
* Layer 6 - Presentation
* Layer 5 - Session
* Layer 4 - Transport
* Layer 3 - Network
* Layer 2 - Data Link
* Layer 1 - Physical

The process of data moving down the layers is called encapsulation. The process of data moving up the layers is called decapsulation.

### Further Research

* [OSI model](https://en.wikipedia.org/wiki/OSI_model)
* [The OSI Model Demystified](https://www.youtube.com/watch?v=HEEnLZV2wGI)
* [The 7 Layers of the OSI Model](https://www.webopedia.com/quick_ref/OSI_Layers.asp)
* [Packet Switching](https://en.wikipedia.org/wiki/Packet_switching)

