

Have you ever wondered how you can troubleshoot a network problem? Or how an email you send from Mauritius is delivered to London in a matter of seconds? The magic behind it is the TCP/IP model. Discover everything you need to know about the backbone of the internet and communication transfer.

### **TCP/IP: What Is the TCP IP Model?**

Computers and the internet have completely transformed the way we communicate. But just like humans, to be able to interact and exchange information successfully, computers need a common way to communicate. Most of the time, they do that through the transmission control protocol and the internet protocol model (TCP/IP). TCP/IP is basically a set of globally standardized rules that enable devices to communicate across networks. No matter which brand they are, where they're located, and what operating system or software they use, TCP/IP works, always.

Have you ever tried buying something in a physical shop, in a foreign country, without speaking the language? I did, and let me tell you, it was an interesting experience. As my knowledge of Japanese was limited to "sushi," "arigato" and "sumo," I tried hand gestures, but it only caused puzzled looks.

To cut the story short, the magic happened when the shop assistant found a colleague who could speak a bit of English (the only one available that day apparently). That day, I not only got a new, shiny, smartwatch, but I also realized how important it is to have a common way to communicate clearly.

### **TCP vs. IP: Two Protocols Working Together as One**

You must have noticed that the two protocols TCP and IP have always been mentioned together. Why? Because even if they're two separate communication protocols, they work together toward a common goal, operating on different TCP/IP stack layers. (We'll speak more about TCP IP layers a little later.) Like the Japanese writer Rynosuke Satoro said, "Individually, we are one drop. Together, we are an ocean."

TCP is responsible for transporting and delivering the data to the destination application or device defined by the IP. While IP is the part obtaining the address to which the data is sent.

It's a bit like when you send a birthday card to your grandmother: the address you write on the envelope is the IP, while the process and the technology that make sure the letter is delivered to the correct address is TCP.

Whether you send an email, browse the internet, or watch your favorite movies, the TCP/IP model makes all of it possible. How? It all starts with a handshake — much like how you shake hands when you meet a new colleague at work or greet a person you know in the street.

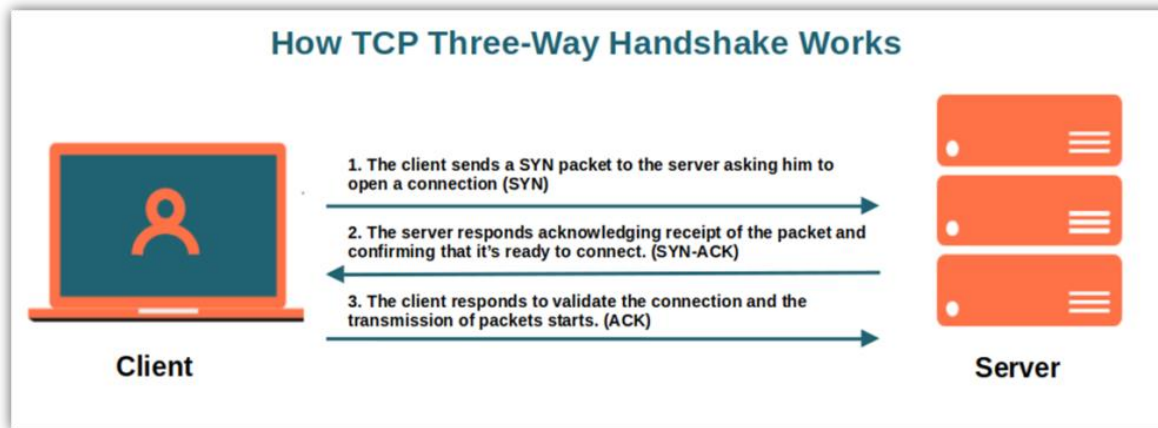
### **TCP and the Three-Way Handshake**

Before transmitting packets, TCP must ensure that a stable connection has been set up between the sender and the recipient. This is where the three-way handshake (or SYN-SYN-ACK) comes in. Much like how you'd traditionally shake hands when meeting a new colleague at work or greeting someone you know on the street, there needs to be a similar exchange of messages and acknowledgments that takes place between the two parties in a TCP connection.

The TCP three-way handshake works as follows:

1. The client sends a SYN packet to the server asking him to open a connection.
2. The server responds by sending a SYN-ACK, acknowledging receipt of the SYN packet, and confirming that it's ready to connect.
3. The client responds with ACK to validate the connection and the transmission of packets starts.

Easy — like shooting fish in a barrel!



A basic overview of the *TCP three-way handshake process*.

With the handshake done and dusted, we can now go deeper and explore the architecture behind the TCP/IP model (i.e., the TCP/IP model layers).

### Breaking Down the TCP/IP Model

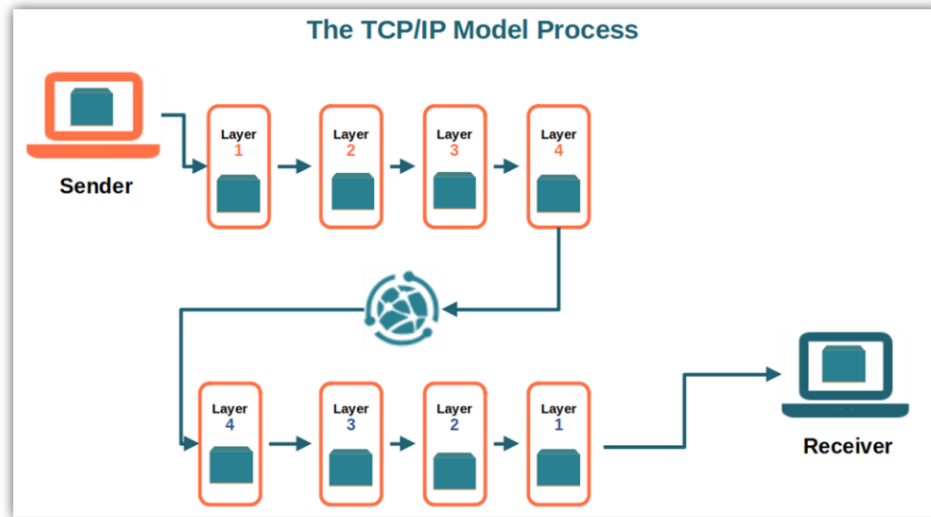
After completing your last big project, you want to take a few days off from work. You open your mail account, write your request to your boss, click on send and keep your fingers crossed for a positive reply. And abracadabra! Your message is suddenly transmitting from your email account to your bosses where, hopefully, they'll quickly approve it.

But what happens between the time you hit "send" and when it arrives in his inbox?

- **The TCP/IP model splits the email into small packets.** As this protocol puts a lot of emphasis on accuracy, the message is broken down into packets. This way, if something happens during the transmission (e.g., a packet is lost), the system will just have to re-send the lost packet, not the whole message. You don't want to end up with your boss receiving only an empty email, right?
- **The packets go through four layers.** Each layer has a different function. The layers define the standards for data exchange and how your email is going to be handled when transmitted.

- **The data are received by the recipient.** To reassemble the message, the packets go through the same four layers in reverse. Now your boss can read your email.

If you're lucky enough, you'll soon be sitting by the pool under the sun with a cool drink in your hand.

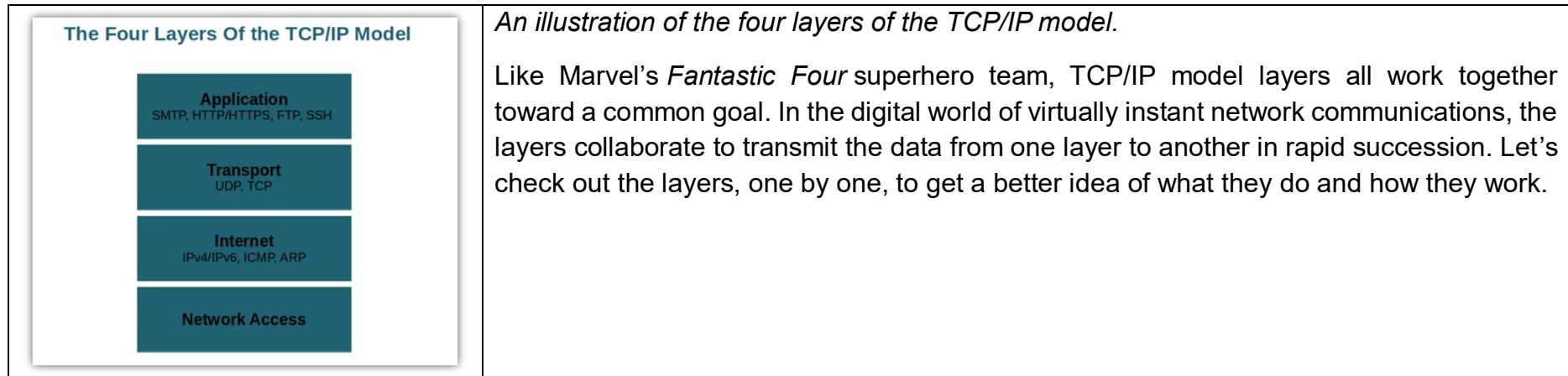


*This graphic illustrates the process of how data flows through the TCP/IP model as it transmits from a sender to the recipient via the internet. Did you notice that the order of the layers flips depending on whether you're the sender or the recipient in this scenario?*

Not yet, though! We still have to clarify the architecture behind the TCP/IP model. For instance, what are those four layers we've just mentioned?

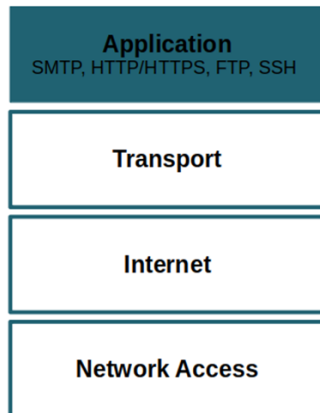
- Application layer
- Transport layer
- Internet layer
- Network Access layer

What do they do? Keep on reading to find it out.



## Layer 1: The Application Layer

### TCP/IP Model: The Application Layer



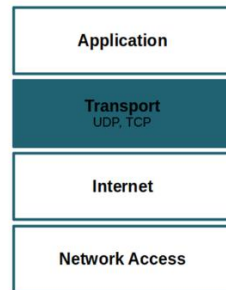
*An illustration that highlights the application layer of the TCP/IP stack.*

The application layer is the highest layer of the model. It's what you, as a user, see and interact with when sending and receiving data. It's also what generates the data and requests the connection. The application layer would be your email system in the previous example of requesting time off from work. Some of the most important protocols used in this layer are:

- [Simple mail transfer protocol \(SMTP\)](#). Used to send emails. This is the protocol you used when you sent the email to your boss.
- [Hypertext transfer protocol/secure \(HTTP/HTTPS\)](#). Both used to access the web. HTTPS (the more secure version of HTTP thanks to encryption) is the protocol you're using to read this article. They're the protocols that you use every time you browse the internet.
- [File transfer protocol \(FTP\)](#). Used to transmit files from one system to another. That's what you would use to upload content if you would have your own website.
- [Secure shell \(SSH\)](#). Similar to Telnet. It's used by system administrators to securely access to access a computer over an insecure network.

## Layer 2: The Transport Layer

#### TCP/IP Model: The Transport Layer



*An illustration that highlights the transport layer within the TCP/IP stack.*

It establishes a reliable and error-free data connection between the application or device and the destination. The transport layer splits the data into smaller packets and numbers them in sequence. It determines:

- The amount of data to be sent,
- The data's destination, and
- The data transmission rate (flow control).

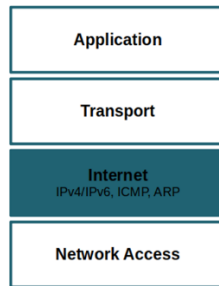
It also obtains the acknowledgment that the recipient has received the packets.

The protocols used in this layer are:

- User datagram protocol (UDP). Used to provide connectionless service and end-to-end delivery of transmission. It identifies transmission errors without specifying them.
- **Transmission control protocol (TCP)**. Used to ensure reliable transmission of packets. It also establishes a network connection between the source and destination.

### Layer 3: The Internet Layer

#### TCP/IP Model: The Internet Layer



*An illustration that highlights the internet layer of the TCP/IP model.*

The internet layer, also called the IP or network layer (though not to be confused with the network access layer, which we'll discuss momentarily), is responsible for sending the packets and ensures that the data is transferred as accurately as possible. It's a bit like the road traffic controller as it directs the flow and speed of traffic. It also provides the functions and procedural methods for the data sequences transfer.

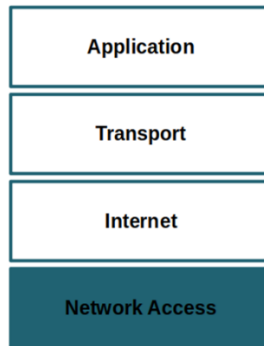
The protocols belonging to this layer are:

- **Internet protocol versions 4 and 6 (IPv4/IPv6)**. Used for routing the data across the network. It's responsible for packets delivery from the host source to the destination host by looking at the IP addresses in the packet.
- **Internet control message protocol (ICMP)**. Used to provide information to the hosts in case of network problems.
- **Address resolution protocol (ARP)**. Used to find the hardware address of a host from a known IP address.

## Layer 4: The Network Access Layer



#### TCP/IP Model: The Network Access Layer



*A basic diagram that shows where the network access layer falls within the TCP/IP model.*

The network access layer, also called the data link layer or physical layer, is the final layer in the TCP/IP model hierarchy. It corresponds to the data link layer and physical layer of the open systems interconnection (OSI) model (we'll talk more about it in a bit). It:

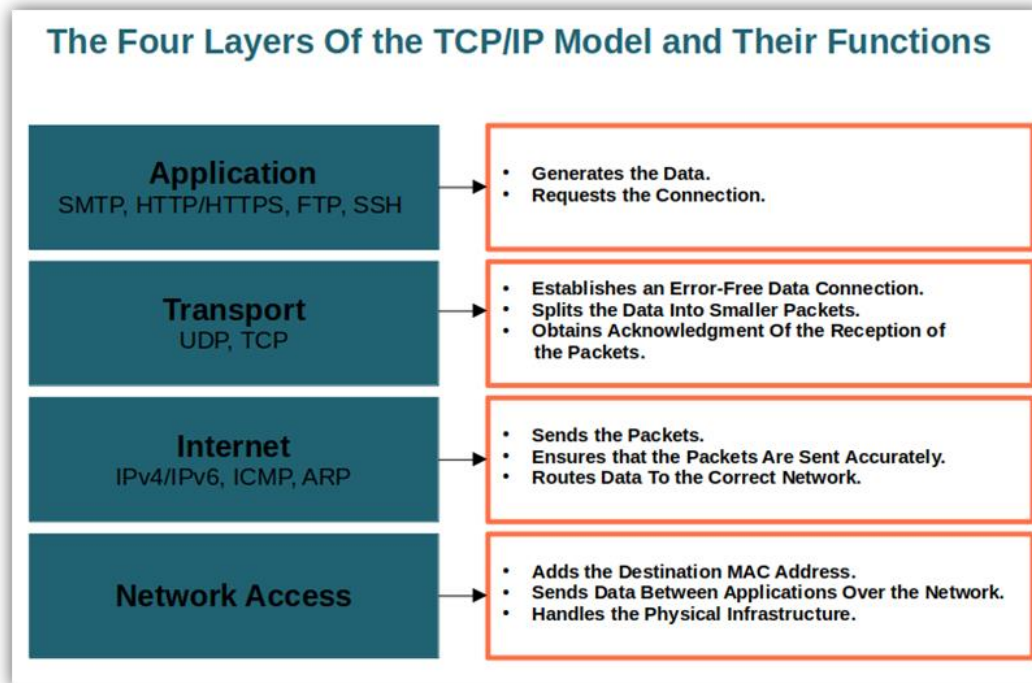
- Adds the destination [MAC address](#) to data frames, and
- transmits data between applications or devices over the network.

It also handles the physical infrastructure enabling communication between devices over the internet. This includes:

- Ethernet cables,
- wireless networks,
- network interface cards,
- drivers, and more.

I know it's a lot to take in. But let's see if the graphic below summarizing the layers and their functions can help you out. I bet it does because often a picture is worth a thousand words.

## The Four Layers Of the TCP/IP Model and Their Functions



*An overview that breaks down the functions and protocols represented by the TCP/IP layers.*

One of the important things to know about the TCP/IP model is that its layers are slightly different from those included in the OSI model we just mentioned. But what's the OSI model? And how is it related to the TCP/IP model? You'll find the answers to these questions in the next chapter.

### TCP Model vs OSI Model — Concise vs Verbose

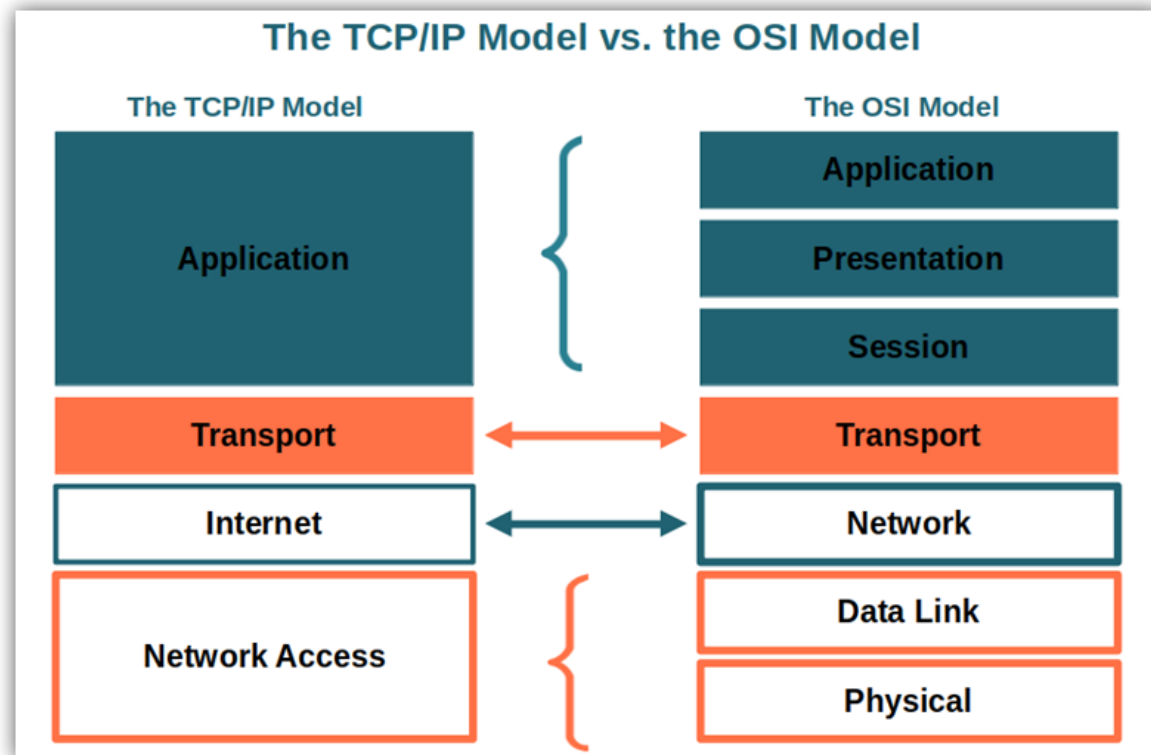
Do you remember when talked about the importance of a common way to communicate? The open systems interconnection model was developed by the International Standards Organization (ISO) to address this need. Its goal? Ensure compatibility between electronic communication systems at a global level.

While the TCP/IP model was an implementation model designed to maintain and explain the transmission of data, OSI was more a kind of reference model. It's a concept that illustrates the functions of the communication system and its procedures. So while both the TCP/IP and OSI models are both about computer networking at their core, there are some fundamental differences between the two models:

	<b>Transmission Control Protocol Model (TCP)</b>	<b>Open Interconnection Systems Model (OSI)</b>
<b>Model Type</b>	Implementation model. Internet is implemented around it.	Reference model. Theoretical.
<b>Number of Layers</b>	Four.	Seven.
<b>Development</b>	The protocol was developed before the model.	The protocol was developed after the model.
<b>Created By</b>	It was developed by the <a href="#">Defense Advanced Research Project Agency</a> (DARPA), part of the <a href="#">U.S. Department of Defense</a> (DOD).	It was developed after the advent of the internet by the ISO.

<b>When Was It Created?</b>	It was created in the 1960s, before the advent of the internet.	It was created in the 1980s, after the advent of the internet.
<b>Layers Structure Difference</b>	The data link layer and physical layer are combined in a single layer called the network access layer. The session and presentation layers are not included in the model.	The data link layer and physical layer are two separate layers. The session and presentation layers are part of the model.
<b>Interfaces, Services, and Protocol Characteristics</b>	There is no clear distinction between services, interfaces, and protocols.	There is a clear distinction between interfaces, services, and protocols.
<b>Minimum Headers Size</b>	20 bytes.	5 bytes.

An Irish proverb says, "Say a little and say it well." This statement embodies the TCP IP model; it's a concise version of the OSI model. As mentioned in the table above, the TCP IP model has four layers, whereas the OSI model has seven layers as described in the graphic below. OSI is, therefore, verbose (extensive). It's a bit like that professor you had in college who could talk for 10 minutes to describe a concept that could be expressed in less than one minute.



A side-by-side comparison illustration of the TCP/IP model *structure* and OSI model *structure*.

## **Final Thoughts on What Is the TCP Model? An Exploration of TCP/IP Layers**

As an IT professional, having a strong understanding of the TCP/IP model and its uses will serve you well in your day-to-day responsibilities. With cyber attacks on the rise, knowing TCP/IP inside and out will enable you to protect your network infrastructure against external threats and optimize its design accordingly.

As an internet user, intermittent network problems and streaming buffering can be frustrating if you don't understand the basic technology behind it. Now that you've had a peek behind the scenes of the TCP model, you'll better be able to point the finger to the right culprit the next time you experience connection issues.

UDP and TCP — they're now familiar terms. But how do you know what's the best internet protocol for you and your organization? Watch this space because this is what we're going to discuss in our next article. We'll be comparing the two protocols and learning about their differences so that you can "get your network to work for you."