

What does TCP/IP mean?

TCP/IP is a suite of [protocols](#). The acronym TCP/IP means "**Transmission Control Protocol/Internet Protocol**" and is pronounced "T-C-P-I-P". It comes from the names of the two major protocols in the suite of protocols, i.e. the [TCP](#) and [IP](#) protocols).

In some ways, TCP/IP represents all communication rules for the internet and is based on the IP addressing notion, i.e. the idea of providing an [IP address](#) for each machine on the network so as to be able to route data packets. Given that the TCP/IP protocol suite was originally created [with a military purpose](#), it is designed to respond to a certain number of criteria, including:

- [Splitting messages into packets](#);
- [Use of an address system](#);
- [Routing data over the network](#);
- [Error detection in data transmissions](#).

Knowledge of the collection of TCP/IP protocols is not essential for a simple user, just as a viewer does not need to know how his television or audiovisual networks work. However, for people who want to administrate or support a TCP/IP network, knowledge of it is vital.

The difference between standard and implementation

Overall TCP/IP brings together two notions:

- The notion of **standard**: TCP/IP represents the way in which communications are carried out over a network.
- The notion of **implementation**: the designation TCP/IP is often extended to software based on the TCP/IP protocol. TCP/IP is in fact a model which network application developers use. The applications are therefore implementations of the TCP/IP protocol.

TCP/IP is a layer model

So as to be able to apply the TCP/IP model to any machine, i.e. independently of the operating system, the TCP/IP system of protocols has been broken down into several modules each carrying out a specific task. Furthermore, these modules carry out these tasks one after the other in a specific order, so there is stratified system, this is the reason we talk about a **layer model**.

The term layer is used to reflect the fact that the data which travels over the network crosses several **protocol levels**. So, data (information packets) which circulates over the network is processed successively by each layer, which adds an item of information (called a *header*) then sends it to the next layer.

The TCP/IP model is very close to the OSI model (model comprising of 7 layers) which was developed by the international standards organisation ([ISO](#)) in order to standardise communications between computers.

Presentation of the OSI model

OSI means *Open Systems Interconnection*. This model was established by ISO in order to implement a communications standard between computers on a network, that is the rules which manage communications between computers. In fact, when networks began, each manufacturer had its own system (we talk of a proprietary system) so many incompatible networks coexisted. This is the why it was necessary to establish a standard.

The role of the OSI model is to standardise communication between machines so that different manufacturers can develop compatible (software or hardware) products (as long as they strictly observe the OSI model).

The importance of a layer system

The aim of a system in layers is to separate the problem into different parts (the layers) depending on their level of abstraction.

Each layer of the model communicates with an adjacent level (above or below it). Each layer therefore uses the services of the lower layers and supplies them to that of the upper layer.

The OSI model

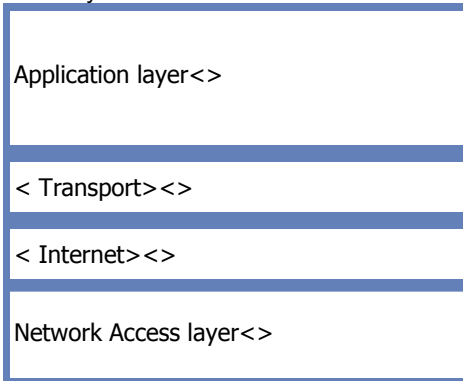
The OSI model is a model which comprises of 7 layers, while the TCP/IP model only has 4. In reality the TCP/IP model was developed around about the same time as the OSI model which is why it is inspired by it but does not completely conform to the specifications of the OSI model. The layers of the OSI model are as follows:

Level	Old model	New model
Level 7<><>		
Level 6<><>		
Level 5<><>		
Level 4<><>		
Level 3<><>		
Level 2< Liaison><>		
Level 1<><>		

- **The physical layer** defines the way in which data is physically converted into digital signals on the communication media (electric pulses, light modulation, etc.).
- **The data link layer** defines the interface with the network interface card and the sharing of the transmission media.
- **The network layer** makes it possible to manage addressing and routing of data, i.e. their path via the network
- **The transport layer** is in charge of data transport, its division into packets and management of potential transmission errors.
- **The session layer** defines the opening and destruction of communication sessions between networked machines.
- **The presentation layer** defines the format of data handled by the application layer (its representation and potentially its compression and encryption) independently of the system.
- **The application layer** provides the interface with applications. It is therefore the closest level to the users, managed directly by the software.

The TCP/IP model

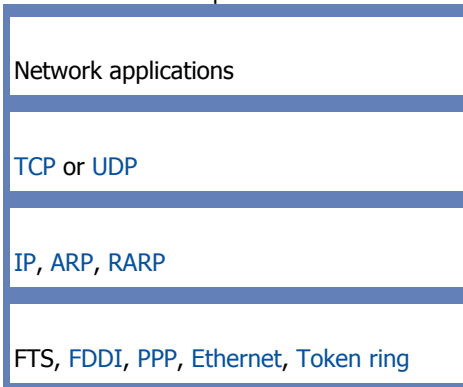
The TCP/IP model, inspired by the OSI model, also uses the modular approach (use of modules or layers) but only contains four:



As can be seen, the layers of the TCP/IP model have much more diverse tasks than those of the OSI model, considering that certain layers from the TCP/IP model correspond to several levels of the OSI model. The roles of the different layers are as follows:

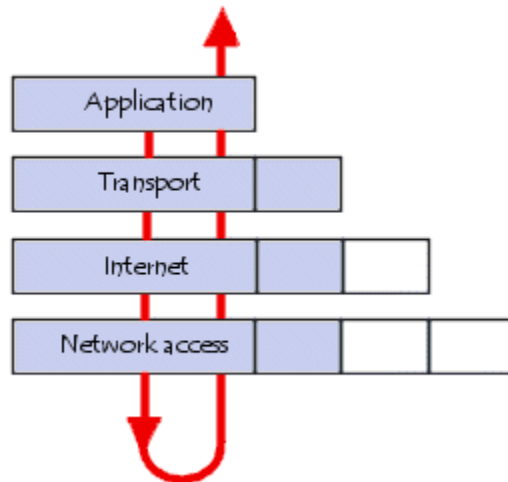
- **Network Access layer:** specifies the form in which data must be routed whichever type of network is used.
- **Internet layer:** responsible for supplying the data packet (datagram)
- **Transport layer:** provides the routing data, along with the mechanisms making it possible to know the status of the transmission
- **Application layer:** incorporates standard network applications (Telnet, SMTP, FTP, ...)

Here are the main protocols which make up the TCP/IP suite:



Data encapsulation

During a transmission, data crosses each one of the layers at the level of the originator machine. At each layer, a piece of information is added to the data packet, this is a **header**, a collection of information which guarantees transmission. At the level of the recipient machine, when passing through each layer, the header is read, then deleted. So, upon its receipt, the message is in its original state...



At each level, the data packet changes aspect, because a header is added to it, so the designations change according to the layers:

- The data packet is called a **message** at Application layer level
- The message is then encapsulated in the form of a **segment** in the Transport layer
- Once the segment is encapsulated in the Internet layer it takes the name of **datagram**
- Finally, we talk about a **frame** at the Network Access layer level

Network Access layer

The network access layer is the first layer of the TCP/IP stack, it offers the ability to access whichever physical network, i.e. the resources to be implemented so as to transmit data via a network. So, the network access layer contains all specifications relating to the transmission of data over a physical network, when it is a local area network ([Token ring](#), [Ethernet](#), [FDDI](#)), connected by telephone line or any other type of link to a network. It deals with the following concepts:

- Routing data over the connection
- Coordination of the data transmission (synchronisation)
- Data format
- Signal conversion (analogue/digital)
- Error detection on arrival

Fortunately all these specifications are transparent in the eyes of the user, because all these tasks are in fact carried out by the operating system, while the hardware drivers allow connection to the network (e.g. network card driver).

The Internet layer

The Internet layer is the "most important" layer (they are all important in their way) because it is this which defines the datagrams and manages the IP addressing notions. It enables the routing of datagrams (data packets) to remote machines along with the management of their division and assembly upon receipt. The Internet layer contains 5 protocols:

- [The IP protocol](#)

- [The ARP protocol](#)
- [The ICMP protocol](#)
- [The RARP protocol](#)
- The IGMP protocol

The first three protocols are the most important protocols for this layer...

The Transport layer

The protocols for the preceding layers make it possible to send information from one machine to another. The transport layer enables applications running on remote machines to communicate. The problem is identifying these applications.

In fact, depending on the machine and its operating system, the application may be a program, task, process...

Furthermore, the name of the application may vary from system to system, that is why a numbering system has been put in place so as to be able to associate an application type with a data type, these identifiers are called [ports](#).

The transport layer contains two protocols enabling two applications to exchange data independently of the type of network taken (i.e. independently of the lower layers), these are the following two protocols:

- [TCP](#), a [connection orientated](#) protocol which provides error detection
- [UDP](#), a [connectionless orientated](#) protocol where error detection is outdated

The Application layer

The application layer is located at the top of the TCP/IP protocol layers. This one contains the network applications which make it possible to communicate using the lower layers.

The software in this layer therefore communicates using one of the two protocols of the layer below (the transport layer), i.e. [TCP](#) or [UDP](#).

There are different types of applications for this layer, but the majority are network services, or applications supplied to the user to provide the interface with the operating system. They can be classed according to the services that they provide:

- File and print management services (transfer)
- Network connection services
- Remote connection services
- Various Internet utilities