

# lamaPLC Communication: TCP / UDP Basic

The Internet protocol suite, commonly known as **TCP/IP**, is a framework for organizing the set of communication protocols used in the Internet and similar computer networks according to functional criteria. The foundational protocols in the suite are the *Transmission Control Protocol (TCP)*, the *User Datagram Protocol (UDP)*, and the *Internet Protocol (IP)*. Early versions of this networking model were known as the *Department of Defense (DoD)* model because the United States Department of Defense funded the research and development through DARPA.

The Internet protocol suite facilitates end-to-end data communication by defining how data should be packetized, addressed, transmitted, routed, and received. This functionality is organized into four abstraction layers, which categorize all related protocols based on each protocol's networking scope. An implementation of these layers for a specific application forms a protocol stack. From the lowest to the highest, the layers are the link layer, which handles communication methods for data confined within a single network segment (link); the internet layer, which enables internetworking between separate networks; the transport layer, responsible for host-to-host communication; and the application layer, which manages process-to-process data exchange for applications. The technical standards that support the Internet protocol suite and its protocols are upheld by the *Internet Engineering Task Force (IETF)*. The Internet protocol suite came before the OSI model, which is a more detailed reference framework for general networking systems.

## TCP/IP

The **TCP/IP** (*Transmission Control Protocol / Internet Protocol*) model is the standard method of data communication on the Internet. It was created by the United States Department of Defense to ensure the accurate and reliable transmission of data between devices. It divides messages into packets to prevent resending the entire message if a problem occurs during transmission. Packets are automatically reassembled once they arrive at their destination. Each packet can take a different route from the source to the destination computer, depending on whether the original route becomes congested or unavailable.

TCP/IP organizes communication tasks into layers, keeping the process standardized without requiring hardware and software providers to manage it themselves. Data packets must pass through four layers before reaching the destination device, and then TCP/IP reverses the process to restore the message to its original format.

As a connection-oriented protocol, TCP establishes and maintains a connection between applications or devices until they complete data exchange. It determines how the original message should be divided into packets, numbers and reassembles the packets, and then forwards them to other devices on the network, such as routers, security gateways, and switches, on their way to the destination. TCP also sends and receives packets from the network layer, manages the transmission of any dropped packets, controls flow, and guarantees that all packets arrive at their destination.

A good example of how this works in practice is when an email is sent using **SMTP** from an email server. To begin the process, the TCP layer in the server divides the message into packets, numbers them, and forwards them to the IP layer, which then transports each packet to the destination email server. When packets arrive, they are handed back to the TCP layer to be reassembled into the

original message format and then handed back to the email server, which delivers the message to a user's email inbox.

TCP/IP uses a three-way handshake to establish a connection between a device and a server, ensuring that multiple TCP socket connections can be transferred both ways at the same time. Both the device and server must synchronize and acknowledge packets before communication starts, then they can negotiate, separate, and transfer TCP socket connections.

## The 4 Layers of the TCP/IP Model

The TCP/IP model explains how devices transmit data to each other and supports communication over networks and long distances. It shows how data is exchanged and organized across networks. The model has four layers that establish standards for data transfer and detail how data is handled and packaged when sent between applications, devices, and servers.

The four layers of the TCP/IP model are as follows:

1. **Datalink layer:** The datalink layer specifies how data should be transmitted, manages the physical process of sending and receiving data, and is responsible for transferring data between applications or devices on a network. This includes defining how hardware and other transmission devices, such as a computer's device driver, an Ethernet cable, a network interface card (NIC), or a wireless network, signal data. It is also known as the link layer, network access layer, network interface layer, or physical layer, and it combines the physical and data link layers of the Open Systems Interconnection (OSI) model, which standardizes communication functions in computing and telecommunications systems.

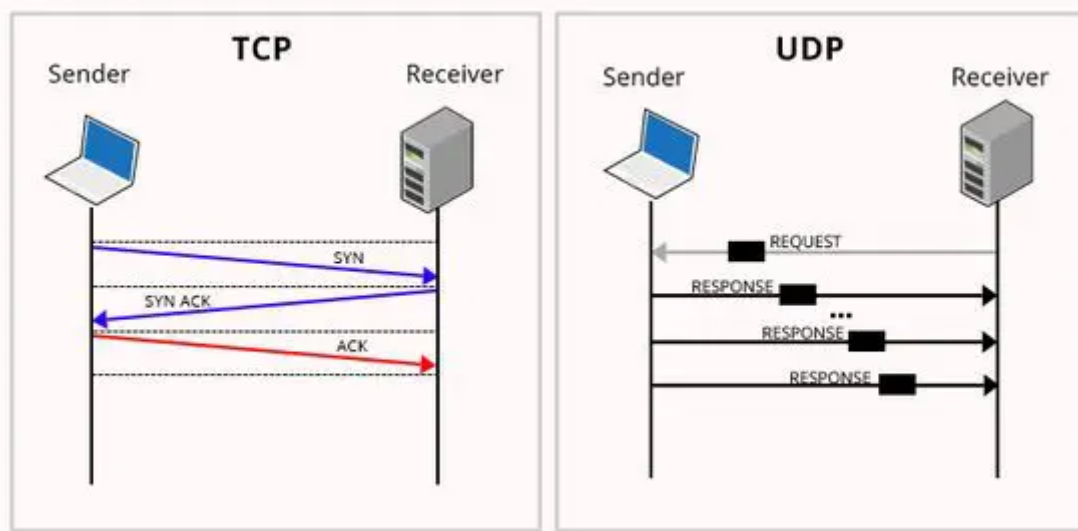
- **Internet layer:** The internet layer handles sending packets across a network and managing their movement to ensure they reach their destination. It provides the functions and procedures necessary for transferring data sequences between applications and devices across networks.

1. **Transport layer:** The transport layer is responsible for providing a solid and reliable data connection between the original application or device and its intended destination. This is the level where data is divided into packets and numbered to create a sequence. The transport layer then determines how much data must be sent, where it should be sent, and at what rate. It ensures that data packets are sent without errors and in order, and it receives acknowledgment that the destination device has received the data packets.
2. **Application layer:** The application layer includes programs that rely on TCP/IP to communicate with each other. This is the level that users normally interact with, such as email systems and messaging platforms. It combines the session, presentation, and application layers of the OSI model.

## TCP & UDP: Differences between the protocols

The main difference between **TCP** (*transmission control protocol*) and **UDP** (*user datagram protocol*) is that TCP is connection-oriented, while UDP is connectionless. Although TCP is more reliable, it transmits data more slowly. UDP is less dependable but operates more quickly. This makes each protocol suitable for different types of data transfers.

Protocols are rules that determine how data is formatted and transmitted over a network. TCP and UDP are two different methods for accomplishing the same task: transferring data across the internet. They allow servers and devices to communicate, enabling you to send emails, stream Netflix, play games, and browse web pages.



TCP establishes a secure communication line to ensure the reliable transfer of all data. Once a message is sent, the receipt is confirmed to verify that all data was transferred.

UDP does not establish a connection when transmitting data. It sends data without confirming receipt or checking for errors. That means some or all of the data might be lost during transmission.

Here are the key differences between TCP and UDP:

Factor	TCP	UDP
Connection type	Requires a established connection before transmitting data	No a connection is needed to start and end data transfer transfer
Data sequence	Can sequence data (send in a specific order)	Cannot sequence or arrange data
Data retransmission	Can retransmit data if packets fail to send arrive	No data retransmitting. Lost data can't be recovered retrieved
Delivery	Delivery is guaranteed	Delivery is not guaranteed
Check for errors	Thorough error-checking guarantees data arrives in its intended state	Minimal error-checking covers the basics but may not prevent all errors
Broadcasting	Not supported	Supported
Speed	Slow, but complete data delivery	Fast, but at risk of incomplete data delivery
Best of..	- Email or texting - File transfers - Web browsing	- Live streaming - Online gaming - Video chat

## TCP Ports

TCP has a feature called a port, which is a unique number assigned to different applications. Standard services (applications) use well-known ports that are documented.

- HTTP → 80
- HTTPS → 443
- FTP → 21
- FTPS / SSH → 22
- POP3 → 110
- Modbus TCP → 502 (503)
- POP3 SSL → 995
- IMAP → 143
- IMAP SSL → 993
- SMTP → 25 (Alternate: 26)
- SMTP SSL → 587
- MySQL → 3306
- cPanel → 2082
- cPanel SSL → 2083
- WHM (Webhost Manager) → 2086
- WHM (Webhost Manager) SSL → 2087
- Webmail → 2095
- Webmail SSL → 2096
- WebDAV/WebDisk → 2077
- WebDAV/WebDisk SSL → 2078
- RSTP → 554
- STP → 128
- unencrypted MQTT → 1883
- encrypted MQTT → 8883

## Sources

Wikipedia ([here](#))

## TCP / UDP topics on lamaPLC

Page	Date	Tags
• <a href="#">lamaPLC Communication: EtherNet/IP</a>	2024/11/17 00:26	<a href="#">communication</a> , <a href="#">ethernet</a> , <a href="#">bus</a> , <a href="#">ip</a> , <a href="#">cip</a> , <a href="#">ethernet ip</a> , <a href="#">udp</a> , <a href="#">industrial ethernet</a> , <a href="#">ie</a> , <a href="#">tcp</a> , <a href="#">unicast</a> , <a href="#">devicenet</a> , <a href="#">controlnet</a> , <a href="#">ieee 802.3</a>
• <a href="#">lamaPLC communication: Matter</a>	2024/11/15 20:21	<a href="#">communication</a> , <a href="#">iot</a> , <a href="#">matter</a> , <a href="#">ethernet</a> , <a href="#">tcp</a> , <a href="#">udp</a> , <a href="#">ipv6</a> , <a href="#">thread</a> , <a href="#">ble</a> , <a href="#">cellular</a> , <a href="#">wifi</a> , <a href="#">pki</a> , <a href="#">ip</a>
• <a href="#">lamaPLC Communication: Modbus</a>	2025/11/19 21:42	<a href="#">modbus</a> , <a href="#">communication</a> , <a href="#">bus</a> , <a href="#">modicon</a> , <a href="#">standard</a> , <a href="#">rtu</a> , <a href="#">tcp</a> , <a href="#">multimaster</a> , <a href="#">coil</a> , <a href="#">register</a>
• <a href="#">lamaPLC Communication: MQTT</a>	2024/11/15 20:28	<a href="#">communication</a> , <a href="#">mqtt</a> , <a href="#">iot</a> , <a href="#">tcp</a> , <a href="#">udp</a> , <a href="#">bluetooth</a> , <a href="#">tcp ip</a> , <a href="#">quic</a> , <a href="#">mqtt-sn</a> , <a href="#">oasis</a> , <a href="#">message broker</a> , <a href="#">mosquitto</a> , <a href="#">qos</a> , <a href="#">arduino</a> , <a href="#">iso iec 20922</a>
• <a href="#">lamaPLC Communication: OPC</a>	2024/11/15 20:33	<a href="#">communication</a> , <a href="#">opc</a> , <a href="#">scada</a> , <a href="#">ole</a> , <a href="#">net</a> , <a href="#">xml</a> , <a href="#">tcp</a> , <a href="#">hmi</a> , <a href="#">server</a> , <a href="#">opc ua</a> , <a href="#">opc ua client</a> , <a href="#">mqtt</a> , <a href="#">json</a> , <a href="#">dcom</a> , <a href="#">simatic</a> , <a href="#">erp</a>
• <a href="#">lamaPLC Communication: PowerLink</a>	2024/11/15 21:15	<a href="#">communication</a> , <a href="#">bus</a> , <a href="#">ethernet</a> , <a href="#">tcp</a> , <a href="#">udp</a> , <a href="#">http</a> , <a href="#">powerlink</a> , <a href="#">fast ethernet</a> , <a href="#">opensafety</a>

- [lamaPLC Communication: TCP / UDP Basic](#) 2025/11/20 21:49 [communication, bus, tcp, udp, tcp ip, darpa, ietf, smtp, http, https, ftp, ftps ssh, pop3, imap, mysql, cpanel, whm, ssl, webmail, mqtt, ethernet, ip](#)
- [lamaPlc: Profinet](#) 2024/11/15 20:39 [communication, bus, profinet, mrp, rstp, gsdml, xml, profibus, ethernet, tcp](#)

[communication, bus, TCP, UDP, TCP/IP, DARPA, IETF, SMTP, HTTP, HTTPS, FTP, FTPS/SSH, POP3, IMAP, MySQL, cPanel, WHM, SSL, Webmail, MQTT, ethernet, IP](#)

This page has been accessed for: Today: 1, Until now: 131

From:  
<http://lamaplc.com/> - **lamaPLC**

Permanent link:  
[http://lamaplc.com/doku.php?id=com:basic\\_tcp](http://lamaplc.com/doku.php?id=com:basic_tcp)

Last update: **2025/11/20 21:49**

