

1 Computer Networks

Jacques Mock Schindler

20.08.2025

Computer networks are systems that enable computers and other devices to communicate with each other. Originally, they were primarily used for exchanging messages. Today, networks also enable the sharing of resources such as printers, files and Internet access. They form the basis for many everyday applications – from e-mails and video conferences to cloud services.

A computer network consists of several interconnected devices (e.g. computers, smartphones, servers) that communicate via various transmission media such as cables or radio waves. Important objectives of a network are:

* Efficient communication * Reliable availability of information * Security of data transmission

Rules and protocols are used in networks to enable communication. A well-known reference model for describing these processes is the Open Systems Interconnection (OSI) layer model. It divides network communication into seven layers that build on top of each other – from the physical transmission of data to the application layer, where programs such as web browsers or email clients operate.

The OSI and the TCP/IP model

The OSI model is a reference model. It helps to structure and understand the complex processes involved in data transmission. The seven layers (with typical data units, known as Protocol or Payload Data Units (PDUs)) at a glance:

1) Physical layer

- Transmits individual bits as electrical, optical or radio signals via the transmission medium (e.g. cable, radio).
- PDU: Bits

2) Data link layer

- Ensures local, frame-based transmission between directly connected devices; regulates access to the medium (medium access control, MAC).
- Provides error detection and, depending on the technology, simple error handling/retransmissions (e.g. for WLAN).
- PDU: Frames

3) Network Layer

- Forwarding packets across multiple networks (routing); addressing at the network level (e.g. IP addresses).
- Example protocol: Internet Protocol (IP).
- PDU: Packets

4) Transport Layer

- Provides end-to-end transport between applications. Different protocols offer different services:
 - TCP: reliable, ordered, connection-oriented; incl. flow and congestion control.
 - UDP: connectionless, no guarantee of reliability or order.
 - PDU: segments (TCP) or datagrams (UDP)

5) Session Layer

- Establishes, manages and terminates sessions between applications (e.g. session management, dialogue control).

6) Presentation Layer

- Translates data between representation formats (e.g. character encoding, serialisation) and can compress/encrypt data.

7) Application Layer

- Provides the interface to application programmes and application protocols (e.g. HTTP, SMTP, DNS).

The OSI layer model has not been implemented as a protocol stack in its pure form. In practice, the TCP/IP model (TCP = Transmission Control Protocol, IP = Internet Protocol) has become established.

The TCP/IP model consists of four layers that combine the functions of the OSI model:

1) Network access layer (OSI: Physical and Data Link Layers)

- Examples: Ethernet, WLAN; MAC addresses; switches typically operate here.
- PDU: Frames

2) Internet layer (OSI: Network Layer)

- Examples: IP, routing; routers operate here; IP addresses.
- PDU: Packets

3) Transport layer (OSI: Transport Layer)

- Examples: TCP (reliable, ordered), UDP (connectionless); ports identify services.
- PDU: Segments (TCP) / Datagrams (UDP)

4) Application layer (OSI: Application, Presentation, and Session Layers)

- Examples: HTTP/HTTPS, SMTP, DNS. Encryption such as TLS is typically above the transport layer and is assigned to applications.

The TCP/IP model is now the basis for communication on the Internet and in most computer networks.

Analogy to encapsulation in object-oriented programming (OOP)

The layer model in network technology can be compared to the principle of encapsulation in object-oriented programming. In both cases, complex tasks are divided into clearly defined, independent units (layers or classes). Each layer in the network model has a precisely defined task and only communicates with the layer directly above or below it – similar to how a class in OOP hides its internal details and only interacts with other classes via defined interfaces.

This encapsulation reduces complexity, makes changes easier to implement and allows the individual components to remain independently maintainable. Just as a class encapsulates its data and methods in OOP, each layer in the layer model encapsulates the details of its function and provides only the necessary services for the next layer.