



Data link layer

The **data link layer**, or **layer 2**, is the second layer of the seven-layer OSI model of computer networking. This layer is the protocol layer that transfers data between nodes on a network segment across the physical layer.^[2] The data link layer provides the functional and procedural means to transfer data between network entities and may also provide the means to detect and possibly correct errors that can occur in the physical layer.

The data link layer is concerned with local delivery of frames between nodes on the same level of the network. Data-link frames, as these protocol data units are called, do not cross the boundaries of a local area network. Inter-network routing and global addressing are higher-layer functions, allowing data-link protocols to focus on local delivery, addressing, and media arbitration. In this way, the data link layer is analogous to a neighborhood traffic cop; it endeavors to arbitrate between parties contending for access to a medium, without concern for their ultimate destination. When devices attempt to use a medium simultaneously, frame collisions occur. Data-link protocols specify how devices detect and recover from such collisions, and may provide mechanisms to reduce or prevent them.

Examples of data link protocols are Ethernet, the IEEE 802.11 WiFi protocols, ATM and Frame Relay. In the Internet Protocol Suite (TCP/IP), the data link layer functionality is contained within the link layer, the lowest layer of the descriptive model, which is assumed to be independent of physical infrastructure.

Function

The data link provides for the transfer of data frames between hosts connected to the physical link. Within the semantics of the OSI network architecture, the protocols of the data link layer respond to service requests from the network layer, and perform their function by issuing service requests to the physical layer. That transfer can be reliable or unreliable; many data link protocols do not have acknowledgments of successful frame reception and acceptance, and some data link protocols might not even perform any check for transmission errors. In those cases, higher-level protocols must provide flow control, error checking, acknowledgments, and retransmission.

The frame header contains the source and destination addresses that indicate which device originated the frame and which device is expected to receive and process it. In contrast to the hierarchical and routable addresses of the network layer, layer 2 addresses are flat, meaning that no part of the address can be used to identify the logical or physical group to which the address belongs.

In some networks, such as IEEE 802 local area networks, the data link layer is described in more detail with media access control (MAC) and logical link control (LLC) sublayers; this means that the IEEE 802.2 LLC protocol can be used with all of the IEEE 802 MAC layers, such as Ethernet, Token Ring, IEEE 802.11, etc., as well as with some non-802 MAC layers such as FDDI. Other data-link-layer protocols, such as HDLC, are specified to include both sublayers, although some other protocols, such as Cisco HDLC, use HDLC's low-level framing as a MAC layer in combination with a different LLC layer. In the ITU-T G.hn standard, which provides a way to create a high-speed (up to 1 Gigabit/s) local area network using existing home wiring (power lines, phone lines and coaxial cables), the data link layer is divided into three sub-layers (application protocol convergence, logical link control and media access control).

Sublayers

The data link layer is often divided into two sublayers: logical link control (LLC) and media access control (MAC).^[3]

Logical link control sublayer

The uppermost sublayer, LLC, multiplexes protocols running at the top of the data link layer, and optionally provides flow control, acknowledgment, and error notification. The LLC provides addressing and control of the data link. It specifies which mechanisms are to be used for addressing stations over the transmission medium and for controlling the data exchanged between the originator and recipient machines.

Media access control sublayer

MAC may refer to the sublayer that determines who is allowed to access the media at any one time (e.g. CSMA/CD). Other times it refers to a frame structure delivered based on MAC addresses inside.

There are generally two forms of media access control: distributed and centralized.^[4] Both of these may be compared to communication between people. In a network made up of people speaking, i.e. a conversation, they will each pause a random amount of time and then attempt to speak again, effectively establishing a long and elaborate game of saying "no, you first".

The Media Access Control sublayer also performs frame synchronization, which determines the start and end of each frame of data in the transmission bitstream. It entails one of several methods: timing-based detection, character counting, byte stuffing, and bit stuffing.

- The time-based approach expects a specified amount of time between frames.
- Character counting tracks the count of remaining characters in the frame header. This method, however, is easily disturbed if this field is corrupted.
- Byte stuffing precedes the frame with a special byte sequence such as DLE STX and succeeds it with DLE ETX. Appearances of DLE (byte value 0x10) have to be escaped with another DLE. The start and stop marks are detected at the receiver and removed as well as the inserted DLE characters.
- Similarly, bit stuffing replaces these start and end marks with flags consisting of a special bit pattern (e.g. a 0, six 1 bits and a 0). Occurrences of this bit pattern in the data to be transmitted are avoided by inserting a bit. To use the example where the flag is 01111110, a 0 is inserted after 5 consecutive 1's in the data stream. The flags and the inserted 0's are removed at the receiving end. This makes for arbitrary long frames and easy synchronization for the recipient. The stuffed bit is added even if the following data bit is 0, which could not be mistaken for a sync sequence, so that the receiver can unambiguously distinguish stuffed bits from normal bits.

Services

The services provided by the data link layer are:

- Encapsulation of network layer data packets into frames
- Frame synchronization
- In the logical link control (LLC) sublayer:
 - Error control (automatic repeat request, ARQ), in addition to ARQ provided by some transport-layer protocols, to forward error correction (FEC) techniques provided on the physical layer, and to error-detection and packet canceling provided at all layers, including the network layer. Data-link-layer error control (i.e. retransmission of erroneous packets) is provided in wireless networks and V.42 telephone network modems, but not in LAN protocols such as Ethernet, since bit errors are so uncommon in short wires. In that case, only error detection and canceling of erroneous packets are provided.
 - Flow control, in addition to the one provided on the transport layer. Data-link-layer flow control is not used in LAN protocols such as Ethernet, but in modems and wireless networks.
- In the medium access control (MAC) sublayer:
 - Multiple access methods for channel-access control, for example CSMA/CD protocols for collision detection and re-transmission in Ethernet bus networks and hub networks, or the CSMA/CA protocol for collision avoidance in wireless networks.
 - Physical addressing (MAC addressing)
 - LAN switching (packet switching), including MAC filtering, Spanning Tree Protocol (STP), Shortest Path Bridging (SPB) and TRILL (TRansparent Interconnection of Lots of Links)
 - Data packet queuing or scheduling
 - Store-and-forward switching or cut-through switching

- [Quality of service \(QoS\) control](#)
- [Virtual LANs \(VLAN\)](#)

Error detection and correction

In addition to framing, the data link layer may also detect and recover from transmission errors. For a receiver to detect transmission errors, the sender must add redundant information as an [error detection code](#) to the frame sent. When the receiver obtains a frame it verifies whether the received error detection code matches a recomputed error detection code.

An error detection code can be defined as a function that computes the r (amount of redundant bits) corresponding to each string of N total number of bits. The simplest error detection code is the [parity bit](#), which allows a receiver to detect transmission errors that have affected a single bit among the transmitted $N + r$ bits. If there are multiple flipped bits then the checking method might not be able to detect this on the receiver side. More advanced methods than parity error detection do exist providing higher grades of quality and features.

A simple example of how this works using [metadata](#) is transmitting the word "HELLO", by encoding each letter as its position in the alphabet. Thus, the letter A is coded as 1, B as 2, and so on as shown in the table on the right. Adding up the resulting numbers yields $8 + 5 + 12 + 12 + 15 = 52$, and $52 + 7 = 59$ calculates the metadata. Finally, the "8 5 12 12 15 59" numbers sequence is transmitted, which the receiver will see on its end if there are no transmission errors. The receiver knows that the last number received is the error-detecting metadata and that all data before is the message, so the receiver can recalculate the above math and if the metadata matches it can be concluded that the data has been received error-free. Though, if the receiver sees something like a "7 5 12 12 15 7" sequence (first element altered by some error), it can run the check by calculating $7 + 5 + 12 + 12 + 15 = 51$ and $51 + 1 = 52$, and discard the received data as defective since 52 does not equal 7.

H	E	L	L	O
8	5	12	12	15

More sophisticated error detection and correction algorithms are designed to reduce the risk that multiple transmission errors in the data would cancel each other out and go undetected. An algorithm that can even detect if the correct bytes are received but out of order is the [cyclic redundancy check](#) or CRC. This algorithm is often used in the data link layer.

Protocol examples

- [ARCnet](#)
- [ATM](#)
- [Cisco Discovery Protocol \(CDP\)](#)
- [Controller Area Network \(CAN\)](#)
- [Econet](#)
- [Ethernet](#)
- [Ethernet Automatic Protection Switching \(EAPS\)](#)
- [Fiber Distributed Data Interface \(FDDI\)](#)
- [Frame Relay](#)
- [High-Level Data Link Control \(HDLC\)](#)
- [IEEE 802.2](#) (provides LLC functions to IEEE 802 MAC layers)
- [IEEE 802.11 wireless LAN](#)
- [I²C](#)
- [LattisNet](#)
- [Link Layer Discovery Protocol \(LLDP\)](#)
- [LocalTalk](#)
- [MIL-STD-1553](#)
- [Multiprotocol Label Switching \(MPLS\)](#)
- [Nortel Discovery Protocol \(NDP\)](#)
- [Point-to-Point Protocol \(PPP\)](#)
- [Profibus](#)
- [SpaceWire](#)

- [Serial Line Internet Protocol \(SLIP\)](#) (obsolete)
- [Split multi-link trunking \(SMLT\)](#)
- [IEEE 802.1aq - Shortest Path Bridging](#)
- [Spanning Tree Protocol](#)
- [StarLan](#)
- [Token Ring](#)
- [TRILL](#) (Transparent Interconnection of Lots of Links)
- [Unidirectional Link Detection \(UDLD\)](#)
- [UNI/O](#)
- [1-Wire](#)
- and most forms of [serial communication](#) e.g. [USB](#), [PCI Express](#).

Relation to the TCP/IP model

In the [Internet Protocol Suite](#) (TCP/IP), OSI's data link layer functionality is contained within its lowest layer, the [link layer](#). The TCP/IP link layer has the operating scope of the link a host is connected to, and only concerns itself with hardware issues to the point of obtaining hardware (MAC) addresses for locating hosts on the link and transmitting data frames onto the link. The link-layer functionality was described in RFC 1122 and is defined differently than the data link layer of OSI, and encompasses all methods that affect the local link.

The TCP/IP model is not a top-down comprehensive design reference for networks. It was formulated for the purpose of illustrating the logical groups and scopes of functions needed in the design of the suite of internetworking protocols of TCP/IP, as needed for the operation of the Internet. In general, direct or strict comparisons of the OSI and TCP/IP models should be avoided, because the layering in TCP/IP is not a principal design criterion and in general, considered to be "harmful" (RFC 3439). In particular, TCP/IP does not dictate a strict hierarchical sequence of encapsulation requirements, as is attributed to OSI protocols.

See also

- [ALOHAnet § ALOHA protocol](#)
- [ODI](#)
- [NDIS](#)
- [SANA-II](#) – Standard [Amiga](#) Networking Architecture, version 2

References

1. "X.225 : Information technology – Open Systems Interconnection – Connection-oriented Session protocol: Protocol specification" (<https://www.itu.int/rec/T-REC-X.225-199511-l/en>). Archived (<https://web.archive.org/web/20210201064044/https://www.itu.int/rec/T-REC-X.225-199511-l/en>) from the original on February 1, 2021. Retrieved March 10, 2023.
2. "What is layer 2, and Why Should You Care?" (<https://web.archive.org/web/20100218075030/http://www.accel-networks.com/blog/2009/09/what-is-layer-2-and-why-should-you-care.html>). accel-networks.com. Archived from the original (<http://www.accel-networks.com/blog/2009/09/what-is-layer-2-and-why-should-you-care.html>) on February 18, 2010. Retrieved September 29, 2009.
3. Regis J. Bates and Donald W. Gregory (2007). *Voice & data communications handbook* (<https://books.google.com/books?id=eq1kRHdyXSUC&pg=PA45>) (5th ed.). McGraw-Hill Professional. p. 45. ISBN 978-0-07-226335-0.
4. Guowang Miao; Guocong Song (2014). *Energy and spectrum efficient wireless network design*. Cambridge University Press. ISBN 978-1107039889.
- S. Tanenbaum, Andrew (2005). *Computer Networks* (https://archive.org/details/isbn_9788177581652) (4th ed.). 482,F.I.E., Patparganj, Delhi 110 092: Dorling Kindersley(India)Pvt. Ltd.,licenses of Pearson Education in South Asia. ISBN 81-7758-165-1.
- Odom, Wendel (2013). *CCENT/CCNA ICND1 100-101, CCENT Official cert guide*. Paul Boger, cisco press. ISBN 978-1-58714-385-4.

External links

- [DataLink layer simulation, written in C# \(http://www.codeproject.com/Articles/57072/DataLink-Simulator\)](http://www.codeproject.com/Articles/57072/DataLink-Simulator)
 - [DataLink Layer, Part 2: Error Detection and Correction \(http://www.cs.gmu.edu/~huangyih/656/error.pdf\)](http://www.cs.gmu.edu/~huangyih/656/error.pdf)
-

Retrieved from "https://en.wikipedia.org/w/index.php?title=Data_link_layer&oldid=1282963375"