

Chapter 2

The Foundation of the Internet : TCP/IP Reference Model

*"The borders of my language define
the borders of my world."*

– Ludwig Wittgenstein, (1889 – 1951)

Spanning the world with its almighty presence, today's Internet connects computers, telephones, entertainment electronics and, in a short time also the household devices and the goods we need for daily life. More and more it penetrates the surface of our lives. To enable these different devices to communicate smoothly and efficiently with each other their communication must follow defined rules – so-called communication protocols. These mold the individual layers of Internet communication determining the tasks, level of abstraction complexity and respective range of functions. By what means and way these specifications are put into practice is, however, not defined by the model but depends on the specific implementation. The TCP/IP reference model thus assumed a concrete form through practical application and builds today, as well as in the foreseeable future, a solid foundation for all of the communication tasks on the Internet.

2.1 Communication Protocol and the Layer Model

Let us first take a closer look at the basis of computer communication. The hardware of a computer network is made up of components that have the task of transmitting information, encoded in the form of bits, from one computer to another. If one wanted to organize computer communication solely on this level it would be like programming a computer in a rudimentary machine language, i.e., only using zeros and ones. It would be virtually impossible to control the required effort and complexity needed to carry out this task. As in computer programming, complex software systems – called network operating systems – were therefore created for the control and use of computer networks. With their help, computer networks can be controlled and implemented in a comfortable way from a higher level of abstraction. These network operating system are based on the idea of handling communication tasks and functions in different degrees of abstraction and complexity. Tasks and

functionalities at the same level of abstraction are bundled together into "layers." Structured one on top of the other, different layers are defined in such a way that with the increasing level of abstraction, communication tasks with different complexity are handled. They are available to the user or computer application via a suitable interface. This type of an approach is also called a **layer model of communication**. The protocols acting on different layers are interlocked via the interface and together form a family of communication protocols (protocol family, protocol suite). The user, but also the majority of application programs communicating over the network to exchange data and offer services, only come into contact with this network operating system. It is only an extremely rare case that contact occurs with the network hardware hidden underneath.

2.1.1 Protocol Families

All of the parties involved must agree to follow common, fixed rules concerning the exchange of information to enable communication – and that not only in the case of digital communication in computer networks. This applies to the language used for communication, as well as to all the codes of behavior, that first make an efficient communication possible. In technical language these codes of behavior are described with the term **communication protocol** or simply, **protocol**

In addition to laying down the format of the information to be exchanged by communication partners, a communication protocol specifies a variety of actions necessary for the transmission of this information. With the development of the first computer networks, hardware was the primary focus and protocol software was viewed as secondary. This strategy has changed radically and today protocol software is highly structured. Instead of providing immense, highly complex and universal network protocols that regulate the entire task range in network communication, the problem of network communication plays out according to the principle of "divide and conquer" (divide et impera). There is a breakdown into a multiple number of manageable sub-problems. (Sub-)protocols focusing on a specific problem are provided in each case in order to deal with these.

These special protocols must work together smoothly and seamlessly. This poses a particular problem that is not to be underestimated in its complexity. In order to ensure this interaction, the development of protocol software is seen as a comprehensive task which is to be accomplished through the availability of an accompanying **family of protocols** (protocol stack, protocol suite). All of the individual protocols are efficiently integrated with each other to solve the overall problem of network communication.

While the various protocol families do in fact share many concepts, they are developed independently from one another as a rule and therefore not compatible. It is, however, possible to implement different protocol families simultaneously and parallel on the computers in a network, allowing them to use the same physical network interface without any resulting interference.

The term "protocol" is normally used here with two different meanings. On one hand, protocol is used to define an abstract interface. Included are all functions and operations that are made available over this interface. On the other hand, the term protocol sums up all of the information formats and their meaning. The definition of the **protocol specification** proceeds most of the time as a combination of specifying texts, images, status transition diagrams, and algorithms in pseudocode. It is necessary that the specifications be precise enough to enable the interoperability of different protocol implementations. Two of the different implementations can then successfully exchange information.

2.1.2 The Layer Model

To support protocol designers in their work, tools and models were developed that finely break down the entire process of network communication, ordering it hierarchically. In this way, clear interfaces are established between the individual levels of the hierarchy. These facilitate the largely independent development and improvement of the individual network protocol located on each layer, simplifying this process as much as possible. The best known variation of this model is the **layer model** (protocol stack) (cf. Fig. 2.3). The entire network communication process is separated into individual layers that are arranged one on top of each other. Each layer addresses a sub-problem of the network communication, with the addition of a new abstraction level of communication. The top layer provides the interface for application programs wishing to exchange information with applications on other computers. On the basis of such a layer model, the protocol designer constructs a complete protocol family, the so-called **protocol stack**, in which the individual protocol solves exactly the tasks addressed on each layer. Principally, in such a layer model the transmission of information from the application program of one computer to the application program of another computer follows a specific organization. The information from the source computer is passed along, processed in parts, from the top to the bottom through the different protocol layers. It is then physically transmitted over the transmission medium to the destination computer over the same protocol layers in reverse order – passed from the bottom to the top and finally transferred to the receiving application (cf. Fig. 2.1).

In the layer model, every layer is responsible for solving a specific part of the numerous tasks that come up within network communication. So that these tasks are carried out correctly, command and control information is created on the side of the transmitting computer, and used on each individual layer of the protocol stack. This is added to the transmitted data (cf. Fig. 2.2). At the receiving computer this supplementary information is read by the protocol software corresponding to each layer and processed further. In this way, the transmitted data can be correctly received at the end.

In accordance with the layer model of network communication, the protocol software of a certain layer k at the receiving computer must receive exactly the information

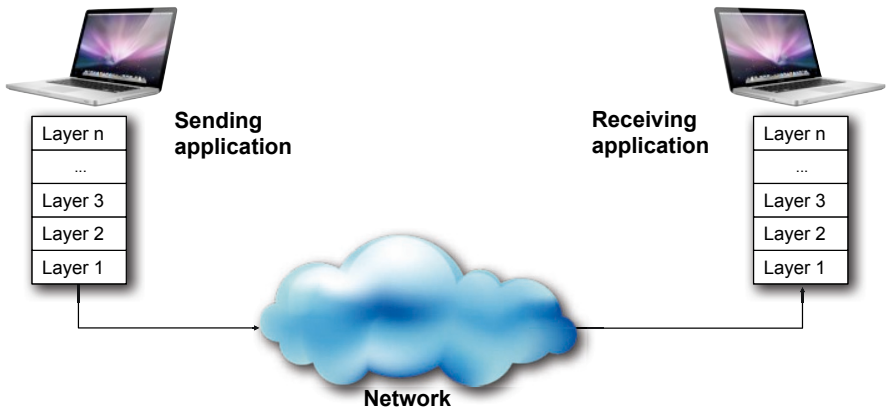


Fig. 2.1 Data transmission via a protocol stack.

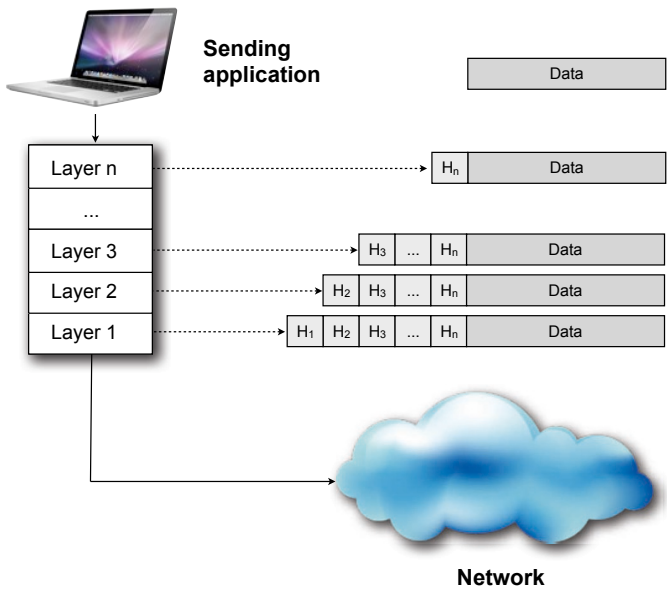
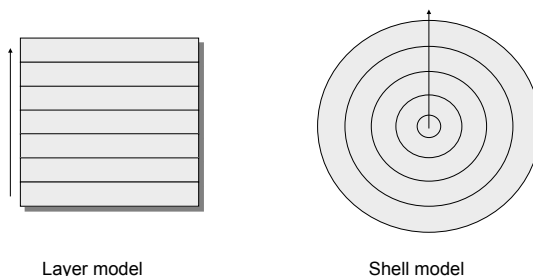


Fig. 2.2 Every layer of the protocol stack adds its own command and control information ($H_n...H_1$) to the data to be transmitted.

General Information about the Layer Model

Layer models play an important role in communication technology as well as in other areas of information science. The **shell model** is a modified form. Rather than being made up of stacked, hierarchically-ordered layers, it is comprised of individual shells.



An application of such a model is useful for the following reasons:

- **Divide and Conquer (Divide et Impera)**

According to this strategy a complex problem is broken down into individual sub-problems. Each one can be handled separately and is thus easier to manage and solve. This might be the only way that a problem can be solved as a whole.

- **Independence**

The individual layers collaborate with each other, every layer only using the interface specifications of its direct predecessor. With fixed, predetermined interface specifications, the internal structure of a layer does not play a role for the other layers. Therefore, implementations on a layer could be exchanged without additional effort in the case of improved implementations. They would only have to be oriented on the respective interface specifications. In this way, implementations at individual layers are **independent** from those at the other layers and a **modular** (building-block) construction of the whole system is made possible.

- **Shielding**

Each individual layer communicates only with its two directly neighboring layers. An **encapsulation** of the single layers is achieved. The challenge posed by the level of complexity to be overcome sinks drastically.

- **Standardization**

The breakdown of the overall problem into individual layers also makes the development of standards easier. An individual layer and its interface allows a faster and easier standardization with the neighboring layers than with the complex system in its entirety.

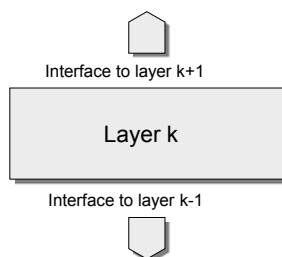


Fig. 2.3 General information about the layer model.

that is sent it by the protocol software of level k at the transmitting computer. This means that every change that the protocol of a specific layer makes to the data being transmitted must be completely reversed by the receiver. If layer k adds additional command and control information to the data to be transmitted, then layer k at the receiving computer must remove it again. If in layer k an encryption of the data has taken place, then at the receiver side the encrypted data in layer k must be decrypted (cf. Fig. 2.4).

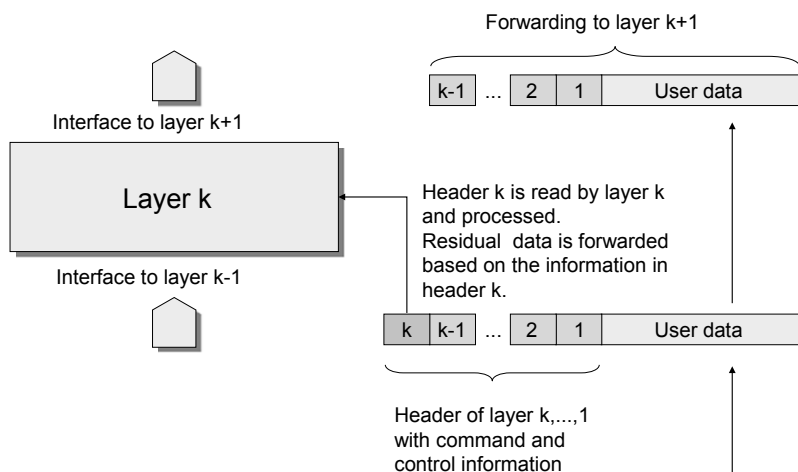


Fig. 2.4 Every layer of the protocol stack reads the data received in the layer-relevant header with the command and control information necessary for processing at this layer.

The actual communication in the protocol stack is always carried out in a vertical direction. When sending data, each protocol layer adds its own command and control information to this data. Typically, this information is passed on to the layer above it in a header prepended to the data packet. In this way the data packet is "encapsulated." The protocol software receives the necessary command and control information from this additional data, which ensures the correct and reliable further processing of transmitted data either on the receiver's side or in the corresponding protocol layer of an intermediate system. At each protocol layer it appears as if the protocol software on both sides – sender and receiver – are in direct communication with each other. But in fact the data is being passed along vertically through the protocol stack. The seemingly direct communication on the individual layers is called **virtual communication** (cf. Fig. 2.5).

Every protocol layer defines two different interfaces. So that applications can use the services of a protocol on its own computer there is a defined **service interface**. The service interface establishes all operations so that local applications can be carried out based on the protocol. Additionally, in every protocol layer there is also a defined

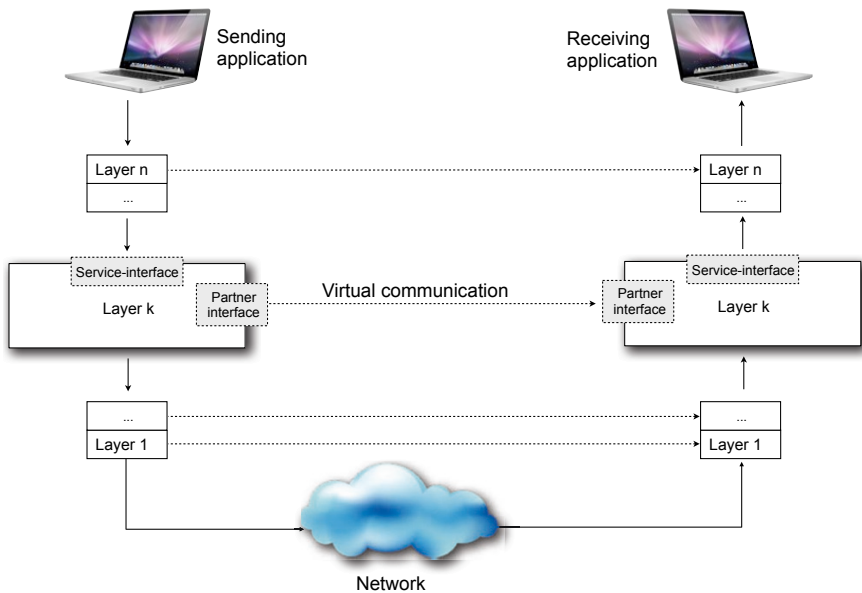


Fig. 2.5 On each level it appears that the individual layers of the protocol stack communicate directly with each other horizontally (**virtual communication**), actually the communication is always carried out vertically.

interface, a counterpart to the corresponding protocol layer on the other computer, the so-called **partner interface** (peer interface). The partner interface specifies the format of information that is exchanged between the neighboring protocol layers on different computers and establishes their meaning. However, communication over the partner interface proceeds in an indirect way. This means that every protocol layer communicates with its counterpart through the delivery of information to a lower, or higher protocol layer, which then sends this information in the same way it to its counterpart at the remote computer.

If a protocol family is implemented in the form of a layer model, several basic areas have to be observed in designing the protocol involved. These can apply to several or to all of the protocol layers So that information can in fact be exchanged between a transmitter and a receiver, a specific form of **addressing** is necessary on every layer so that the correct receiver can be identified among the many that are possible. Furthermore, rules for **data transfer** have to be established at every layer. Does data flow in both directions (bi-directional, duplex operation) or is data traffic only possible in one direction (uni-directional, simplex operation)? Can several (logical) channels be established and used within a communication connection, e.g., a channel for regular data, a channel to control communication, and a further channel for data with high priority? Any error occurring in transmission must also be detected and corrected (**error control**). This task is relevant for all layers and is carried out at different layers with different procedures. Based on technical and logical param-

ters, information to be transferred on individual layers is broken down into smaller sub-units (**fragmentation**).

As the adherence to a specific sequence cannot be guaranteed in every layer, the individual sub-units have to be given a unique **identification**. For example, they are provided with numeration making it possible to put the original information back together at the end. Another problem to be solved involves preventing an especially fast sender at a layer from flooding a slow receiver with information. Here, different methods of **flow control** are carried out to ensure an even network utilization. Individual connections between the transmitter and receiver can be pooled together in the layers above or below or separated again as the case may be. This **multiplexing** (or demultiplexing) must proceed in a transparent way at every layer. If multiplexing takes place in a lower layer then the layer above it should not be influenced. If multiple alternative connection paths exist between the transmitter and the receiver in a network then **routing** decisions must be made. These determine which sections should be chosen in a network for the current processing of information.

In layer models, the successive layers are characterized by a increasing degree of abstraction the higher they become. Data packets are transmitted in the layers near the hardware, while in layers located higher in the protocol stack, information is sent that is broken down into data packets (fragmented) by the protocol software. These higher located layers conceal unnecessary communication details from users and provide comfortable **services** for communication and data transmission.

A general distinction in service is made between **connectionless service** and **connection-oriented service**. Connectionless service works in a way similar to the traditional postal system. Every piece of information is like a letter or package affixed with a complete receiver's address and sent through the network, independent of any other message. As connectionless services do not take prescribed paths through the network, the sequence of the received information packets to the receiver can deviate from that of the sender's. A **reliable service** always confirms the successful delivery of a message by way of an acknowledgment. The transmitter is thus provided certainty as to whether the receiver has in fact gotten the transmitted information. In the case of a **non-reliable service** an acknowledgment of receipt is not provided.

A **connection-oriented service**, in contrast, works in a similar way to the telephone. This means that before a message can be transmitted, a connection to the receiver must be set up. All of the information is then sent along this connection, until the connection is terminated by both communication partners again. Reliable connection-oriented services can transmit data as **message sequences**. In this case, strict attention is paid that message borders are retained upon delivery. As an alternative, reliable, connection-oriented services can also send messages as **byte streams**. Here, message borders are ignored. A further variation is non-reliable, connection-oriented services. Here, a connection is set up prior to data transfer, but transmitter and receiver do not acknowledge confirmation of a message that has been received. This variation is chosen, e.g., for the transmission of audio or video data, as transmission delay caused by the absence of a confirmation receipt would not be acceptable. Transmission errors that occur are perceived as interference or

noise and are sooner tolerated in, e.g., live transmission than delay would be. A **connectionless service** is known as a **datagram service**. In an analogy to the traditional postal service, a datagram service parallels a telegram or a postcard: upon successful delivery the sender receives no acknowledgement confirming this fact. If the (connectionless) communication between the transmitter and the receiver is limited to the exchange of a single message, one speaks of a **request-reply service** (request-reply service). A reliable, connectionless service (**reliable datagram service**), is advantageous when only a short message is to be sent, without the wish to establish an explicit connection. This variation can be compared to a registered letter with receipt. Here, with the receipt of a delivery confirmation the transmitter can be certain that the recipient has in fact really gotten the letter. Table 2.1 presents an overview of these different types of services.

Table 2.1 Service types

	Service	Example
Connection-oriented	reliable message stream	sequence of individual images
	reliable byte stream	terminal login
	unreliable connection	video stream
Connectionless	unreliable datagram	unconfirmed email message
	confirmed datagram	confirmed email message
	request/answer	client/server handshake

The difference between **protocols** and **services** has a special significance in this context. While protocols determine the rules and data formats governing the data that is to be exchanged within a given layer, the service designates a collection of operations (service primitives) that one layer makes available to the layer above it (cf. Fig. 2.5). Service primitives are understood as individual operations that prompt a certain action or report on its status (cf. Fig. 2.6). The specification of a service is summarized in the **service primitive**. However, not the actual manner of how this service is to be carried out is specified, but only the interface description between two neighboring layers in the protocol stack. Protocols, in contrast, carry out the services available in each layer.

Based on the layer model, two important **reference models** have been developed: the ISO/OSI reference model and the TCP/IP reference model. A reference model is a term used to describe an abstract model that serves as the basis upon which concrete implementation can be created. Specific protocols are linked with the two reference models named. These are located on the individual layers of the model

Service Primitives for the Implementation of a Connection-Oriented Service

To implement a connection-oriented service it is necessary that service primitives are available for the following operations:

- **Connection setup:**
In order to establish a connection with a communication partner (on the same layer in the protocol stack), an operation is necessary that takes as parameter the address of the communication partner to which it sends a login (CONNECT).
- **Waiting for connection:**
When one communication partner is ready to connect with another communication partner the latter is put into a special state of waiting for the establishment of the connection (LISTEN).
- **Transmission of messages:**
When the connection has been established the active communication partner can send its partner a message (SEND).
- **Receiving of messages:**
With the establishment of a connection each communication partner alternatively is put in a special waiting state for the message of its counterpart (RECEIVE).
- **Connection termination:**
After the communication is finished a corresponding operation is necessary to actively end the connection to a communication partner (DISCONNECT).

A distinction is made between the active communication partner, who begins a communication process (client) and the passive communication partner (server), who waits for the establishment of communication to it, in order, e.g., to ask for services or information. A server is initially in the LISTEN state and waits for a connection setup. The client starts its request with a CONNECT to a specific server and waits for an answer. If the server receives the connection request, it confirms this and executes RECEIVE to wait for data sent by the client. After receiving the confirmation of a connection establishment, the client executes SEND to send out further requests for services or data. In subsequently executing RECEIVE it waits for the answer from the server. A dialog exchange thereby arises between the client and the server. It is actively ended by the client with the execution of DISCONNECT. The server then also implements DISCONNECT and moves into the LISTEN status again (cf. also the section 8.1.4).

Fig. 2.6 Example for an application by service primitives

and enable those services to be carried out that are contained within that layer. The ISO/OSI model and accompanying protocol were conceived as a theoretical model and had an important didactic significance demonstrating the tasks and services of each layer. Conversely, the TCP/IP reference model evolved from the practical development of the Internet. It was based on a protocol model with practical application.

Excursus 1: ISO/OSI Reference Model

To enable the development of network protocol families, the International Standards Organization (ISO) made the **ISO/OSI Reference Model** available starting in 1977 for communication in open networks (Open Systems Interconnection). It broke down the entire process of network communication into seven individual layers and is designed as a conceptual tool for the development of protocol families (cf. Fig. 2.7).

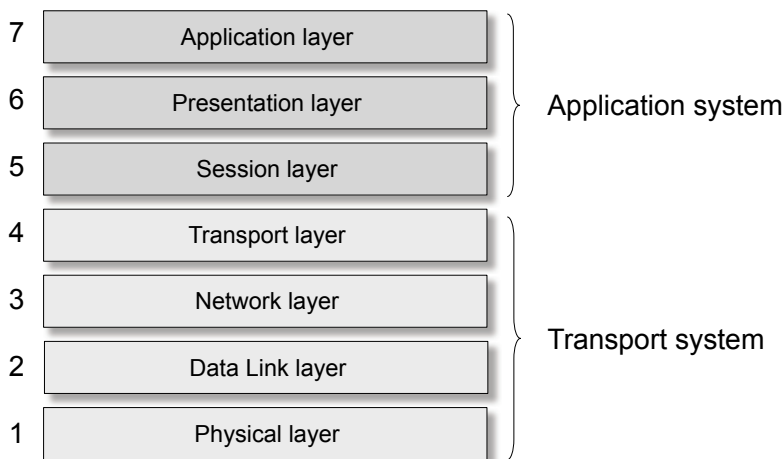


Fig. 2.7 The individual layers of the ISO/OSI Reference Model.

The network protocols existing before the ISO/OSI initiative were mainly of a proprietary nature and developed by the individual manufactures of the network devices themselves. Among these pre-ISO/OSI network protocol standards were, for example, IBM SNA, AppleTalk, Novell Netware and DECnet – all of which were not compatible with each other. While the standardization efforts for the ISO/OSI were still in progress the Internet -based protocol family TCP/IP was rapidly gaining in importance in heterogeneous networks that were comprised of components from different manufacturers. It gained a leading position before the ISO/OS succeeded in achieving standardization.

The ISO/OSI reference model was given the name **Open Systems Interconnection**, as it was intended for the connection of open systems, i.e., systems that are open for communication with other systems. The primary idea behind the design of the ISO/OSI Reference Model was that every single layer was to implement an exactly defined function. A new, higher layer would always be added if a new degree of abstraction was necessary to carry out the tasks at hand. The ISO/OSI model does not itself offer a network architecture. Only the tasks of the individual layers are determined in it and no decisions are made about the implementation of the functionality of the services and protocols.

In the ISO/OSI reference model the bottom layer in the protocol stack corresponds to the actual network hardware (physical level). The layers building on it each comprise the firmware and software implemented on this network hardware. The highest layer (layer seven) is finally the application layer providing an interface between the communication system and the various applications wishing to use the communication system for their own purpose. Layers 1-4 are designated in general as the **transport system** and layers 5-7 as the **application system**. These provide the increasingly general functionalities of the communication process. Although the name is the same, they should not be confused with the actual application programs located outside of the layer model. The tasks of the individual layers of the ISO/OSI reference model are described as follows.

- **Layer 1: Physical layer**

The physical layer defines the physical and technical properties of the transmission medium (transmission channel). In particular, the relation between the network hardware and the physical transmission medium is regulated. For example, layout and assignment

of plug connections with their optical and electrical parameters, cable specifications, amplification elements, network adapters, implemented transmission methods, etc.

Among the most important tasks of the physical layer are:

- Establishment and termination of a connection to a transmission medium and
- Modulation, i.e., conversion of a binary data (bit stream) into (electrical, optical or radio) signals, which can be transmitted over a communication channel.

Important protocol standards of this layer are, e.g.,

- ITU-T V.24, V.34, V.35
- ITU-T X.21 and X.21bis
- T1, E1
- SONET, SDH (Synchronous Data Hierarchy), DSL (Digital Subscriber Line)
- EIA/TIA RS-232-C
- IEEE 802.11 PHY

● Layer 2: Data link layer

In contrast to the physical layer, whose main task is regulating the communication between a single network component and the transmission medium, the data link layer is concerned with the interaction of multiple (at least two) network components. The data link layer ensures that a reliable transmission can take place at a point-to-point-connection – despite potential periodic errors on the physical layer. This point-to-point-connection can be carried out either as a direct connection or via a **diffusion network** as in the case of e.g., Ethernet or WLAN. All connected computers in a diffusion network can receive the transmitted data of all other connected computers without the need for an intermediate system.

The tasks to be carried out on the data link layer include

- the organization of data into logical units, called **frames** on the data link layer,
- the transmission of frames between network components,
- bit-stuffing, i.e., the padding of a frame that is not filled completely with special fill data, and
- the reliable transmission of frames through simple error detection methods, such as checksum calculation.

Among the most well known protocol standards of this layer are:

- BSC (Bit Synchronous Communication) und DDCMP (Digital Data Communications Message Protocol), PPP (Point-to-Point Protocol)
- IEEE 802.3 (Ethernet)
- HDLC (High-level Data Link Protocol)
- X.25 LAPB (Link Access Procedure for Balanced Mode) und LAPD (Link Access Procedure for D-Channels)
- IEEE 802.11 MAC (Medium Access Control)/LLC (Logical Link Control)
- ATM (Asynchronous Transfer Mode), FDDI (Fiber Distributed Data Interface), Frame Relay

- **Layer 3: Network layer**

The network layer provides the functional and procedural means to enable the transfer of data sequences of variable lengths (**data packets**) from a transmitter to a receiver over one or more networks.

Numbering among the tasks of the network layer are:

- the assigning of addresses to end and intermediate systems,
- the targeted forwarding of data packets from one end of the network to the other (**routing**) and thereby
- linking individual networks (**internetworking**),
- the fragmentation and reassembly of data packets, as different networks are determined by different transport parameters, and
- the forwarding of error and status notification, related to the successful delivery of data packets.

Some of the most important protocol standards at this layer are:

- ITU-T X.25 PLP (Packet Layer Protocol)
- ISO/IEC 8208, ISO/IEC 8878
- Novell IPX (Internetwork Packet Exchange)
- IP (Internet Protocol)

- **Layer 4: Transport layer**

The transport layer facilitates a transparent data transfer between end users, providing the layers above it with a reliable transport service. The details necessary to ensure a reliable and secure data transmission are defined by the transport layer. Here it is assured that a sequence of error-free and complete data packets reach the receiver from the transmitter. On the transport layer the mapping of network addresses to logical names take place. The transport layer provides the participating end systems an end-to-end connection. Because it conceals the details of the network infrastructure in between it is described as **transparent**. The protocols located on this layer number among the most complex in network communication.

Among the most important protocol standards in layer 4 are:

- ISO/IEC 8072 (Transport Service Definition)
- ISO/IEC 8073 (Connection Oriented Transport Protocol)
- ITU-T T.80 (Network-Independent Basic Transport Service for Telematic Services)
- TCP (Transmission Control Protocol), UDP (User Datagram Protocol), RTP (Real-time Transport Protocol)

- **Layer 5: Session layer**

The session layer is also called the communication control layer because it controls the dialog between two computers connected over the network.

The main duties of the session layer include:

- the establishment, management and termination of all connections between local and distance services,
- the controlling of full-duplex or simplex data transport and
- the establishment of security mechanisms, for example authentication via the password procedure.

Important protocols of this layer are:

- SAP (Session Announcement Protocol), SIP (Session Initiation Protocol)
- NetBIOS (Network Basic Input/Output System)
- ISO 8326 (Basic Connection Oriented Session Service Definition)
- ISO 8327 (Basic Connection Oriented Session Protocol Definition)
- ITU-T T.62 (Control Procedures for Teletex and Group 4 Facsimile Services)

- **Layer 6: Presentation layer**

The presentation layer creates a context between two entities (applications) of the application layer above it, so that the two applications are able to use different syntax (e.g., data formats and encoding) and semantics. The presentation layer is therefore responsible for a correct interpretation of the transmitted data. Additionally, the respective local encoding of the data is transcribed in a special, standardized transfer encoding and transformed back at the receiver into the locally valid encoding .

Additionally, data compression and encryption belong to the tasks on this layer.

Among the most important protocol standards of the presentation layer are:

- ISO 8322 (Connection Oriented Session Service Definition)
- ISO 8323 (Connection Oriented Session Protocol Definition)
- ITU-T T.73 (Document Interchange Protocol for Telematic Services), ITU-T X.409 (Presentation Syntax and Notation)
- MIME (Multipurpose Internet Mail Extensions), XDR (External Data Representation)
- SSL (Secure Socket Layer), TLS (Transport Layer Security)

- **Layer 7: Application layer**

The application layer provides an interface for application programs wishing to use the network for their own purposes. Application programs themselves do not belong in this layer but only use its services. The application layer provides simple and easy-to-manage service primitives that conceal network internal details from the user or the programmer of the application program and therefore enable a simple use of the communication system. Some of the most important functions of the application layer are:

- identification of the communication partner,
- determination of the availability of resources and the
- synchronization of communication.

Numbering among the most important protocol standards located on this layer are:

- ISO 8571 (FTAM, File Transfer, Access and Management)
- ISO 8831 (JTM, Job Transfer and Manipulation)
- ISO 9040 und 9041 (VT, Virtual Terminal Protocol)
- ISO 10021 (MOTIS, Message Oriented Text Interchange System)
- FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), HTTP (Hypertext Transfer Protocol), etc.
- ITU-T X.400 (Data Communication for Message Handling Systems). ITU-T X.500 (Electronic Directory Services)

Since the development of the ISO/OSI Reference Model, concepts for protocol families in different locations have changed and many of the newly developed protocols no longer fit into this scheme. Nevertheless, a large part of the terminology, especially designations and numeration of the individual layers, has remained the same until today.

Further Reading:

U. Black: OSI – A Model for Computer Communications Standards, Upper Saddle River, NJ, USA (1991)

H. Zimmermann: OSI Reference Model – The ISO Model of Architecture for Open Systems Interconnection, in IEEE Transactions on Communications, vol. 28, no. 4, pp. 425–432 (1980)

2.2 The Physical Layer as the Basis for Computer Communication

The protocols in the lowest layer of the TCP/IP Reference Model (the network access layer) are based on the physical transmission medium (transmission channel). This transmission medium is also called the **physical layer**, but it is normally not included in the protocol stack of the TCP/IP reference model (cf. Fig. 2.10). The physical layer together with the four layers of the TCP/IP reference model comprise the so-called hybrid TCP/IP reference model. In contrast, the physical layer forms a layer of its own with the same name in the ISO/OSI reference model .

2.2.1 Physical Transmission Media

In general, the physical layer defines the physical and technical properties of a physical or analog transmission medium, used for data transmission. The relations between the network hardware and the physical transmission medium are particularly regulated, such as the layout and assignment of plug connections with their optical/electrical parameters, cable specifications, amplification elements, network adapters, implemented transmission methods, etc. The actual task of the physical layer is to translate a series of bits (bit stream) into a sequence of physical signals, which is then forwarded from the sender to the receiver with the help of the transmission medium.

Depending on the nature of the transmission medium, numerous methods and processes can be used. Bit sequences are transformed securely and reliably into physical signals in different ways to be sent over the transmission medium. On the side of the receiver they are reassembled again into the output information. This procedure is known as **modulation**, while in the opposite direction it is called **demodulation**. Media implemented for data transmission can generally be separated into the groups of **wired** (guided) transmission media and **wireless** (unguided) transmission media. With wired (guided) transmission media, electromagnetic waves are forwarded along a solid medium. There are many different variations of this medium, from copper cable, such as twisted pairs or coaxial cable (wirepath, conductor), to different fiber optical cable variations (fiber optics) (light path, waveguide). To gain

access to a network based on a wired transmission medium, a direct, physical contact must first be created. The high transmission speed results from the low rate of error, which can be achieved thanks to good shielding possibilities. However, wired network architectures also involve substantial costs as cables must be bought and laid.

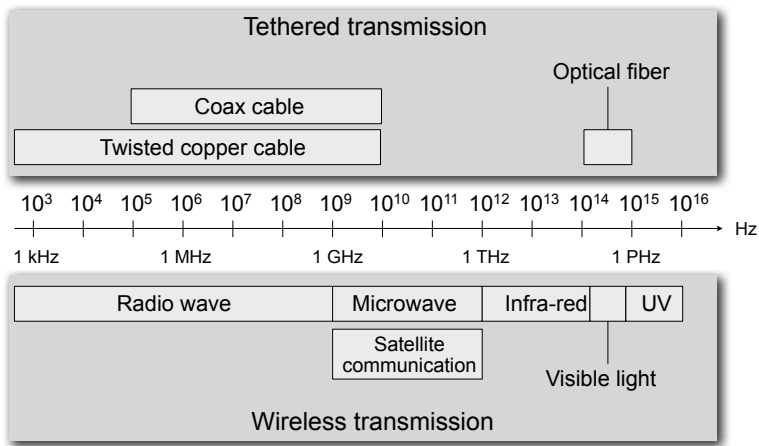


Fig. 2.8 Wired and wireless transmission media in the electromagnetic spectrum.

In the case of wireless (unguided) transmission media, electromagnetic waves are transmitted over different frequency ranges of the electromagnetic spectrum into space. These include radio transmission via short or ultra-short wave, microwave transmission, infrared or laser light. A distinction is made between directed transmission, such as with a laser beam, directional radio or satellite direct radio link, and undirected (isotropic) transmission, such as mobile communication, terrestrial or satellite broadcast. In comparison to wired transmission media, a wireless network architecture is flexible and ideal for mobile implementation. There are no costs for complex cabling. But, on the other hand, it is possible to penetrate a wireless network without direct physical contact being necessary. For this reason, the use of complex software technologies are a necessary security measure, for example encrypted data transmission. Furthermore, transmission speed is decreased due to signal transmission disturbances caused by reflections from objects or atmospheric interference.

2.2.2 Characteristic Properties of Physical Transmission Media

All physical transmission media are restricted by their individual limitations. This affects the maximum information (bandwidth) transported per time unit or the speed

at which a signal can spread over the transmission medium. Generally, every signal that spreads along a physical medium is subject to a signal damping. As the distance to the transmitter increases, the signal weakens accordingly. In contrast to a perfect transmission medium, actual transmission media is constantly at the mercy of interference (noise). If signal damping weakens the signal to such an extent that it can no longer be distinguished from noise, then the signal on the side of the receiver cannot be reconstructed or correctly interpreted. For this reason, as a signal moves along the transmission medium it must be refreshed in certain places, i.e., strengthened, in order that it can be received with the greatest reliability and in the purest form possible.

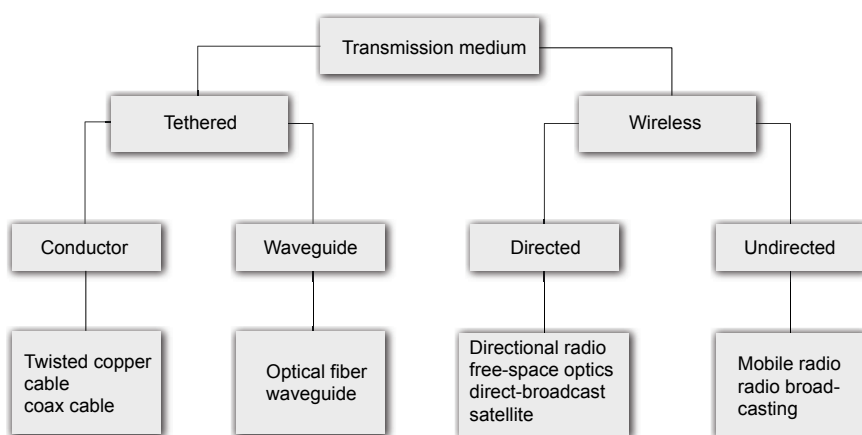


Fig. 2.9 A classification of physical transmission media.

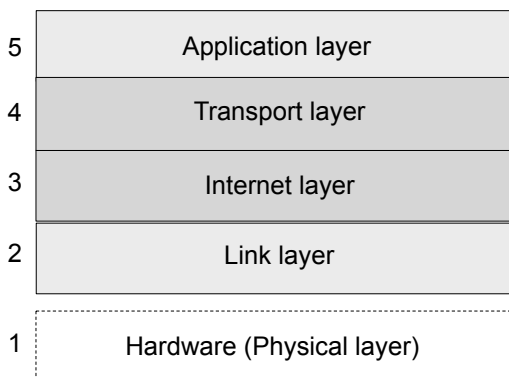
Independent of the physical transmission medium, different modulation procedures are implemented to ensure that the encoding and transmission of binary (digital) information over a physical (analog) medium is as efficient as possible. The various physical transmission media, their limitations, characteristic properties as well as their employment in the Internet will be looked at in detail in Chap. 3.

2.3 The TCP/IP Reference Model

The complex tasks that are necessary for computer communication on the Internet are regulated with the help of different hierarchically linked protocols. Their functionality is best described with a **layer model**, as was presented in the last chapter. A group of tasks is defined in each layer that must be handled by the protocols assigned to that layer. The protocols of any given layer must take into account the neighboring layer above and below it. The interface to the adjacent layers on the same terminal

is called the **service interface**. Over the service interface, a protocol provides a specific service to a layer higher in the protocol stack and in carrying out its task can fall back on the services of the layers below. In addition, a protocol defines a further interface to its counterpart on a remote terminal in the form and meaning of the information exchanged between the terminals. This interface is therefore also called a **partner interface**. Communication within the layer model proceeds in a vertical direction between neighboring layers, with the actual data exchange only occurring on the lowest level, the physical layer. In contrast, via the partner interface a seemingly virtual connection is implemented over which data and control information is exchanged. In each case only a specific protocol layer is affected. At the time of the Internet's development the focus was on making seamless communication possible over a multiple number of network architectures. Based on both of the primary Internet protocols: the Internet Protocol (IP) and the Transmission Control Protocol (TCP), the resulting architecture was designated the TCP/IP reference model. The actual TCP/IP reference model encompasses four layers (layers 2–5). Those together with the addition of the physical layer (layer 1) make up the five-layer hybrid TCP/IP reference model (cf. Fig. 2.10).

Fig. 2.10 The TCP/IP reference Model is comprised of four layers (2-5), together with the physical layer (1). This model is also called the hybrid TCP/IP reference model.



The **link layer** of the TCP/IP reference model corresponds to the first two layers of the ISO/OSI reference model (physical layer and data link layer). Its main task focuses on the secure transmission of data packets of pooled bit sequences. It is followed by the **internet layer**, which corresponds to the network layer of the ISO/OSI reference model. Its main responsibility is to enable the data communication of two end systems at a given location in the heterogeneous communication network. The **transport layer** above it corresponds to the layer of the same name in the ISO/OSI reference model. It enables two user programs on different computers in the communication network to exchange reliable and connection-oriented data. The **application layer** of the TCP/IP reference model includes the three upper layers of the ISO/OSI reference model and serves as an interface for the actual application programs that wish to communicate with each other over the network.

The **TCP/IP reference model** stands in marked contrast to the ISO/OSI reference model. Unlike the ISO/OSI reference model, it was not conceived and planned theoretically but was derived from the protocols that had been put into practice on the Internet.

The ISO/OSI protocols, on the other hand, were planned theoretically and adopted before protocols could be invented that implemented the different function for the layers of the ISO/OSI reference model. Today these protocols are no longer used and the TCP/IP reference model protocols, which had been developed from practical application, dominate the Internet. After a short excursion into the historical development of the TCP/IP protocols, a look will be taken at the similarities and differences between the TCP/IP reference model and the ISO/OSI reference model. Finally, the individual layers of the TCP/IP reference model will be presented.

2.3.1 Historical Background and Distinction from the ISO/OSI Reference Model

The Internet, and its successor the ARPANET, came into use purely as research networks starting in 1969.

The initial four computers connected to the network increased to several hundred in just a short time, linking universities, research institutes and military installations with each other over leased telephone lines. As technologically different networks, such as satellite networks and wireless networks, were connected to the Internet the originally implemented protocols were soon overburdened by the data traffic translation required from one network to the other.

In 1972, *Robert E. Kahn* (*1938) was working on data transmission in satellite networks and wireless networks at the DARPA Information Processing Technology Office (IPTO), which was responsible for the further development of the Internet at that time. There, he soon realized the advantage of enabling data traffic via different network technologies. A new, flexible network paradigm had to be developed with the initial focus on linking heterogeneous networks based on different technologies. *Vinton Cerf* (*1943) joined Kahn's team in 1973. He had been one of the developers of the **Network Control Program**, at that time still implemented as a network protocol in ARPANET, and he joined Kahn to work on protocols for an open network architecture

In the summer of 1973, Kahn and Cerf introduced a fundamentally new network architecture. Its main characteristic was to virtually unite the different network technologies over a common "Internetworking protocol," via the respective protocols of the actual network technologies. In contrast to the existing ARPANET, where the network itself was responsible for reliable data transport, from now on end systems (hosts) connected with networks should be responsible for reliable data transmission. The functionality of the network itself would be limited from this time on to the simplest possible data packet transport. In making this move, Kahn and Cerf also succeeded in connecting the most different network technologies with one another.

The connection of different networks was to proceed via special computers, so-called **package brokers** (routers), which are solely responsible for the forwarding of data packets between different networks. Cerf's own research group at Stanford University worked on the first specification of the **Transmission Control Protocol** (TCP, RFC 675) until 1974.

They were strongly influenced by the network research group of the Xerox PARC (Palo Alto Research Center), and the work being done there on the PARC Universal Packet Protocol Suite (PARC UPPS).

The DARPA commissioned the company BN Technologies, Stanford University and University College London with the technical implementation. Their task was to employ the new protocol standard on different hardware platforms. Following the versions TCP v1 and TCP v2, the protocol was broken down and further developed into TCP v3 and IPv3. The resulting network architecture was later given a designation based on the two most important protocols, TCP and IP (Internet Protocol), as TCP/IP reference model (RFC 1122). This led to the 1978 development of an operational version **TCP/IP v4** (Version 4), which is still used on the Internet today. proof of the operational readiness of the TCP/IP could be demonstrated with the linking of two different networks between Stanford University and the University College London via TCP/IP. A test with three different network architectures followed between the USA, Great Britain and Norway in 1977. The final transition of the complete Internet to TCP/IP v4 took place on January 1, 1983.

Robert E. Kahn and Vinton Cerf received the highest award in information technology for their achievements in 2004 – the Turing Award. A year later both were recipients of the the Presidential Medal of Freedom, the highest civil order of merit in the USA. At the 2nd German IPv6 Summit in 2009 at Hasso Plattner Institute in Potsdam, Robert E. Kahn was named a HPI Fellow, an honor he shares with German Chancellor Dr. Angela Merkel and, since 2011, with his former colleague, Vinton Cerf.

Both the ARPANET as well as the Internet already existed when the ISO tackled the development and standardization of the ISO/OS reference model. The TCP/IP reference model manifested on the Internet, therefore had a decisive influence on the development of the ISO/OSI reference model. The seven layers of the ISO/OSI reference model allow themselves to be transferred to the protocol architecture of the Internet, whereby the TCP/IP reference model is only made up of four layers (or five layers in the hybrid TCP/IP reference model, with the inclusion of the physical layer). Implementation possibilities and application areas (network access, Internet, transport, application) are specified in the individual layers of the TCP/IP reference model. In the ISO/OSI reference model, concrete rules are given for operation, semantics of the data and network technologies. The TCP/IP reference model does not contain concrete hardware specifications and neither does it standardize the physical data transmission as such, but rather binds these aspects to the implementation of individual layers.

The most important protocol family today, the **TCP/IP** protocol suite, is not based on the specifications of a standardization committee, but grew out of requirements and experience from the developing Internet. The ISO/OSI reference model is ad-

aptable to the extent that it can also serve to describe the TCP/IP protocol stack, but both express different principles.

The **TCP/IP reference model** was in fact first defined completely after the protocols described in it were implemented and being used successfully. This had the advantage that the described layer specifications corresponded perfectly with the protocol implementation. However, an application of this model in other protocol families could not be carried out easily. The first description of the TCP/IP reference model (RFC 1122) can already be found as early as 1974, even before the first specifications of the ISO/OSI model were carried out.

Principally, the TCP/IP protocol family can be divided into four single layers, which are organized around the core layers TCP and IP (cf. Fig. 2.10). In fact descriptions of the TCP/IP reference model as comprised of five different layers can also be found in technological literature. The communications hardware descriptive layer (physical layer, hardware) is included in the original four layer TCP/IP reference model. This five layer model is often called the **hybrid TCP/IP Reference Model**. The designation of the single layers correspond to the underlying RFC 1122 and will be used throughout this book.

The four layers of the TCP/IP Reference Model can be compared in the following way with the seven layers of the ISO/OSI Reference Model (cf. also Fig. 2.11):

- Layer 2 of the TCP/IP reference model (link layer) is often designated as the data link layer in technological literature, or is also called the network access layer or host-to-network layer. It corresponds to the first two layers of the ISO/OSI reference model (physical layer, data link layer).
- Layer 3 of the TCP/IP reference model (Internet layer) is also called the network layer or Internetwork layer and corresponds to layer 3 of the ISO/OSI reference model (network layer).
- Layer 4 of the TCP/I reference model (transport layer) is also designated the host-to-host layer and corresponds to layer 4 of the ISO/OSI reference model (transport layer).
- Layer 5 of the TCP/IP reference model (application layer) corresponds to layers 5 – 7 of the ISO/OSI reference model (session layer, presentation layer, application layer).

In the following sections, the tasks and protocols of the individual layers of the TCP/IP reference model will be looked at in more detail.

2.3.2 *Link Layer*

The link layer of the TCP/IP reference Model combines the first two layers of the ISO/ISO reference model: layer 1 – the physical layer and layer 2 – the data link layer. However, the link layer does not contain the aspects of the physical layer that are part of the ISO/OSI reference model. The link layer is therefore the lowest

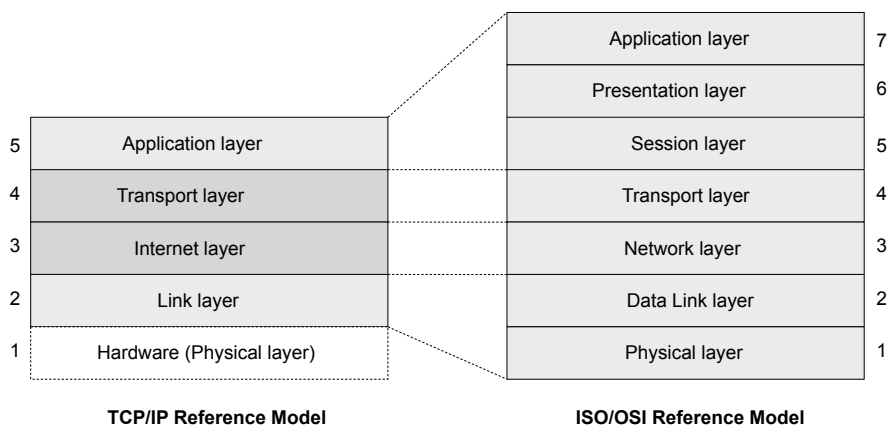


Fig. 2.11 A comparison of the TCP/IP Reference Model and the ISO/OSI-Reference Model.

layer of the TCP/IP reference model. The primary task of the link layer is the secure transmission of individual data packets between two adjacent end systems. The bit sequences to be transmitted are pooled together into fixed units and provided with the additional information necessary for transmission, e.g., checksums for simple error detection. The adjacent end systems can be either directly connected with each other by a transmission medium or by a so-called bus (diffusion network), which connects multiple end systems directly, therefore without intermediate systems.

On this layer a distinction is made between **secured** and **unsecured services**. With unsecured services, data packets recognized as defective are eliminated. The request for the necessary re-transmission follows, but first on a higher layer of the protocol stack. In contrast, a secured service takes over the request for a retransmission itself.

In local networks (LANs), layer 2 of the TCP/IP reference model is normally subdivided into two further layers:

- **Medium Access Control (MAC)**

This sublayer regulates access to the shared (with many other) computer systems transmission medium. As these are in competition with each other to gain access to the transmission medium, it is necessary that protocol mechanisms are provided allowing fair and efficient access to all participants (**multiple access protocols**). This includes methods for the discovery of collisions or their avoidance, as many participants wish to transmit data at the same time (**collision detection, collision avoidance**). Additionally, every participant at this layer must have an individual and unique address as a means of identification (**MAC addressing**). On the MAC sublayer different (but homogeneous) subnets are already connected with each other over a so-called **switch (LAN switching)**. Thereby data packets are each forwarded only within the subnet where the respective target computer is located. Here, the switch takes over the task of filtering the data traffic (**MAC filtering**). Two types of switches may be identified. The **store-and-forward** switch

always save the data packets to be filtered before they are analyzed and finally forwarded. The **cut-through** switch carries out the forwarding without prior caching. Additionally, in this sublayer there are also tasks of administering (**data packet queuing** and **scheduling**) if data packets cannot be forwarded quickly enough before new data packets are delivered and it is necessary to determine priority.

- **Logical Link Control (LLC)**

This sublayer forms the so-called data link layer of the LAN. The tasks regulated there are on a higher abstraction level than the MAC sublayer below upon which it is located. Problems and tasks of the LLC sublayer are determined in the **IEEE 802.2** standard. Among its tasks are avoiding overload situations by potential receivers of transmitted data through targeted interventions in the data flow (**flow control**) and control of data transmission (**link management**). Over the LLC sublayer a first quality control of the transmitted data also takes place. Data transmission errors must be recognized and if possible corrected. For this purpose the protocol located on the LLC sublayer carries out various **error detection and error correction procedures**. Additionally, the LLC sublayer synchronizes the transmitting and receiving of data units (data packets). To enable this, the data must be subdivided into size-limited data packets (**fragmentation**) in accordance with the physical and logical conditions of the respectively chosen transmission form. Following transmission it is essential that the beginning and ending of a data packet always be correctly recognized (**data packet synchronization**). Besides this, the LLC sublayer ensures so-called **multi-protocol capability**. This is the capability of using different communication protocols at the same time.

Included with the most important protocols of the link layer of the TCP/IP reference model are those from the IEEE, based on the **IEEE 802** LAN standard of the standardized LAN protocols. These are technologies such as **Ethernet** (IEEE 802.3), **token ring** and **FDDI** (IEEE 802.5), as well as different wireless **WLAN** technologies (IEEE 802.11). We will take a closer look at these in Chap. 4.

The most important protocols of the link layer of the TCP/IP protocol family are:

- **ATM (Asynchronous Transfer Mode)**

ATM is a packet-switching network protocol that breaks down and forwards the data to be transported in cells of a fixed size (cell relay). Behind this design principle was the idea of providing time-critical, real-time data, such as video or audio information, together with regular data over a standardized protocol. Attention was paid to keeping switching and transfer delay as small as possible. ATM is connection-oriented and additionally establishes a virtual connection between two endpoints in the network before the actual data transfer begins. ATM is employed in LAN as well as in wide area networks, so called WANs.

- **ARP (Address Resolution Protocol) and RARP (Reverse Address Resolution Protocol)**

With the help of the ARP protocol described in RFC 826 the MAC address of a host can be determined from the IP address in the Internet protocol of the layer

above it. This is important if a data packet from the Internet is to be delivered to a local network whose MAC address must be determined from the stored IP address of the receiver for forwarding in the LAN. ARP is only implemented in LANs or point-to-point connections. The reverse service is delivered by the standardized RARP protocol in RFC 903. For each MAC address a corresponding IP address can be determined .

- **NDP (Neighbor Discovery Protocol)**

The functions of the NDP protocols are very similar to the ARP protocol and serve to explore and discover further hosts in the local network. In contrast to ARP, NDP was developed for the next generation of the Internet protocol IPv6, while ARP works under the current version IPv4.

- **LLTD (Link Layer Topology Discovery)**

The proprietary LLTD protocol was developed by the Microsoft company for exploration of the present network topology and verification of the guaranteed quality of service in a network.

- **SLIP (Serial Line Interface Protocol) and PLIP (Parallel Line Interface Protocol)**

SLIP, established in RFC 1055, and PLIP are simple point-to-point network protocols that serve in the transport of encapsulated IP data packets between personal computers over a serial (SLIP) or parallel interfaces, i.e., these IP data packets are packed in data packets of the SLIP or PLIP protocols. For the most part, SLIP and PLIP have been replaced by modern PPP protocol.

- **PPP (Point to Point Protocol)**

PPP is a simple point-to-point network protocol that serves the connection between two network nodes. PPP is used by the majority of Internet providers (ISPs) to offer their customers a dial-up connection over a standard telephone line in the Internet. Modern access via DSL (Digital Subscriber Line) is implemented by the ISPs over the encapsulated protocol variations PPPoE (PPP over Ethernet, RFC 2516) and PPPoA (PPP over ATM, RFC 2364).

- **STP (Spanning Tree Protocol)**

STP was established in the IEEE 802.1D standard, and describes a protocol that is intended to ensure cycle freedom within a LAN architecture consisting of multiple network segments. As indicated in the name of the protocol, a so-called spanning tree is created from the network graph that is present. Over the spanning tree it is guaranteed that the LAN does not contain any closed loops, where the data packets would then be traveling for an unlimited time.

2.3.3 Internet Layer

The main task of the TCP/IP reference model Internet layer involves enabling data communication between two end systems on different ends of the communication network, i.e., over different network architectures.

The methods described in the Internet layer for bridging and uniting different network architectures with the help of special intermediate systems (routers) is also called **Internetworking**. The necessary tasks to be solved are described in RFC 1122. In Internetworking there needs to be a clear addressing scheme that exceeds the given network borders (**IP addressing**). Each data packet to be sent has to be provided with the addresses from the transmitter and receiver in order to be delivered correctly. As the communication proceeds over one or more independent operating networks, the computers at the connection and switching centers (intermediate systems, routers) must be in a position to choose the correct path to follow when forwarding data packets (**routing**). In data packet switching across different network types, often different rules are valid in connection with the maximum size of data to be transported in a single data packet (Maximum Transmission Unit, MTU). The transmitting intermediate systems therefore have to break down data packets (**fragmentation**) to be transmitted in a network with stricter limitations. At the receiver they will then be put back together.¹ Furthermore, further technical differences can occur between the bridged networks. These have to be compensated by appropriate transfer and calculation methods, for example the switching between encrypted networks and unencrypted networks or different time or volume-based calculation procedures.

The three basic tasks of the Internet layer are composed of the following functions:

- Outgoing data packets must be forwarded to the next switching location or to the receiving end system. To do this, the responsible communication protocol must choose the next (direct) receiver (**next hop**) along the mediated path and send the data packet to it through transfer to the respective, responsible protocol in the link layer.
- Incoming data packets have to be unpacked, the control information read from the header of the data packet and, if relevant, the transported user data passed on to the active transport protocol in the above layer.
- Additionally, diagnostic tasks are taken over and a simple error handling implemented. However, on the Internet layer there are just **unreliable services** offered, i.e. there is no guarantee that a transmitted data packet actually reaches its receiver. The transport can therefore take place only "as well as possible" (**best effort**). Control of a reliable communication is the responsibility of the two end-points of communication (transmitter and receiver). It is carried out on a higher layer of the TCP/IP reference model to unburden the network on this layer from this difficult task. The scalability and error tolerance of Internet technology is first made possible through this "best effort" strategy. Only in this way has it been possible for the Internet to grow to its present-day size.

The central protocol of the Internet layer is the **Internet Protocol (IP)**. IP offers an unreliable, data packet-oriented, end-to-end information transmission. It is responsible for fragmentation and defragmentation into so-called **IP datagrams** and has

¹ This task is omitted in the new version of the Internet protocol IPv6. The end systems involved in the communication carry out a pre-fragmentation themselves in order to facilitate a faster transmission of data.

protocol mechanisms for forwarding via intermediate systems to the designated receiver of the information. IP exists today in two versions, IPv4 (RFC 791) and IPv6 (RFC 2460). It takes its place among the most important protocols in the Internet. In addition, the protocol **ICMP (Internet Control Message Protocol)** is implemented in the Internet layer. It is responsible for the notification of specific errors occurring during IP transmission, as well as for further diagnostic tasks, such as sending echo requests to test the availability of a computer and the necessary transmission time. ICMP is a protocol that sits directly on the IP. Two different variations of the ICMP protocols exist, one for IPv4 (RFC 792) and another for IPv6 (RFC 4443).

Besides IP and ICMP there are further protocols that number among those in the Internet layer of the TCP/IP protocol stack, for example:

- **IPsec (Internet Protocol Security)**

IPsec is comprised of a protocol suite to ensure the secure execution of IP data traffic. Within a data stream IP datagrams can be authenticated (Authentication Header, AH) and encrypted (Encapsulating Security Payload, ESP) (RFC 4835). Additionally, included in IPsec are protocols for the handling, establishment and exchange of secure cryptographic keys (Internet Key Exchange Protocol, IKE, RFC 2409).

- **IGMP (Internet Group Management Protocol)**

The IGMP protocol (RFC 1112, RFC 2236, RFC 3376) carries out the administration of IP multicast groups of end systems within a TCP/IP network. Special multicast routers administer address lists of end systems that can be addressed commonly via one multicast address. Through the use of multicast addresses the burden on the transmitter and on the entire network is reduced. IGMP exists only in one version for IPv4, as IPv6 implements multicasting differently.

- **OSPF (Open Shortest Path First)**

The OSPF protocol (RFC 2328) is a so-called link-state routing protocol, that transmits IP datagrams within a single routing domain (autonomous system). It belongs to the group of the Interior Gateway Protocols (IGP). OSPF is the most widely used routing protocol on the Internet.

- **ST 2+ (Internet Stream Protocol, Version 2)**

The Internet stream protocol (ST, RFC 1190, und ST 2+, RFC 1819) is an experimental protocol of the Internet layer. As a supplement to the Internet protocol, it is intended to provide a connection-oriented transport of real-time data with the guarantee of a constant quality of service.

2.3.4 Transport Layer

The transport layer in the TCP/IP reference model corresponds roughly to layer 4 of the ISO/OSI reference model. Its primary task entails the establishment and use of a communication connection between two application programs residing on different computers in the network. The protocol of the transport layer establishes a

direct, virtual end-to-end communication connection. To allow multiple application programs on the same computer to have parallel communication, a statistical multiplexing is among the tasks of the transport layer. Every application program is assigned a so-called **port number** to provide unique identification. On the transport layer, every unit of data sent must contain the port number of the sender and the receiver in order to be transmitted correctly. Together with the IP address, the port number defines what is commonly called a **network socket**, a unique connection endpoint in the network. A complex flow control is likewise implemented on the transport layer. This ensures that overload situations are avoided to the greatest extent possible (congestion avoidance). Finally, measures are taken to ensure that the transmitted data arrives at the receiver error-free and in the correct order (sequence numbers). An acknowledgment mechanism is provided over which the receiver can confirm correctly transmitted data packets, or send a new request in the case of defective data packets.

Unlike the Internet layer, the transport layer is not under the control of the network operator. It therefore offers the user or application program of the communicating end systems the possibility to influence problems in data transmission that are not handled at the Internet layer. These include the bridging of failures on the Internet layer and the subsequent delivery of data packets that have gotten lost on the Internet layer.

The **Transport Control Protocol (TCP)** is a further core element of the Internet protocol architecture and the most popular protocol of the transport layer in the TCP/IP reference model. Standardized as RFC 793, it carries out a reliable, connection-oriented, bi-directional data exchange between two end systems, which in the TCP/IP reference model is based on an unreliable, connectionless datagram service of the Internet layer. TCP enables the establishment of what are called **virtual networks** (virtual circuits). After a virtual connection is set up, a data stream (byte stream) is transmitted that hides the packet-oriented transmission of information from the application layer above it. A reliable service is thereby initiated by an acknowledgment mechanism (Automatic Repeat Request, ARQ), over which the transmission of lost data is initiated.

Next to TCP, the **Universal Datagram Protocol (UDP)** is the second most prominent protocol of the transport layer. Standardized as RFC 768, it transmits independent data units, so-called datagrams, between application programs that reside on different computers in networks. However, the transmission is unreliable, i.e., possibly combined with data loss, proliferation of datagrams and changes in sequence. The datagrams recognized as false are discarded by UDP and do not even reach the receiver. In comparison to TCP, UDP is clearly less complex, which is reflected in its increased data throughput. Yet this is compensated by a dramatic loss of reliability and security. The application program above has to take care of this itself.

Other important protocols of the transport layer are:

- **DCCP (Datagram Congestion Control Protocol)**

DDCP (RFC 4340) is a message-oriented protocol of the transport layer. In addition to reliably establishing and terminating connections, it also distributes overload notifications (Explicit Congestion Notification, ECN). It provides overload

control functions (congestion control) and can be used for the negotiation of transmission parameters.

- **RSVP (Resource Reservation Protocol)**

The RSVP (RFC 2205) protocol is used for the request and reservation of network resources using IP to transmit data streams. It is not intended for the actual data transport and bears a similarity to the ICMP and IGMP protocols at the Internet layer. RSVP can be implemented by end systems as well as routers without having to reserve and maintain specified service qualities.

- **TLS (Transport Layer Security)**

As predecessor of the **Secure Socket Layer Protocol (SSL)**, TLS supplies (RFC 2246, 4346 and RFC 5246) cryptographic protocols for secure data transport in Internet. Individual TCP segments are encrypted by TLS and SSL.

TLS provides protocols for the negotiation of transmission parameters (peer negotiation), for the exchange of cryptographic keys and authentication as well as for encrypting and digital signature.

- **SCTP (Stream Control Transmission Protocol)**

SCTP (RFC 4960) is a proposal for a highly scalable and performant version of the original TCP protocol and is specialized in transmitting large amounts of data.

2.3.5 Application Layer

The functions on the application layer in the TCP/IP reference model encompass the tasks of layers 5-7 in the ISO/OSI reference model. The application layer primarily functions as an interface for the actual application programs wishing to communicate over the network (process-to-process communication). The applications themselves are located outside of this layer and even outside of the TCP/IP reference model.

The services offered and programming interfaces (Application Programming Interface, API) of the application layer have a high level of abstraction. The user or communicating application is for the most part shielded from the details of communication, which is regulated on the lower protocol layers. Protocols and services of the application layer normally carry out translations and transformation of data between application programs at the semantic level. Among the services offered are: naming services, which translate IP addresses into readable names and visa versa, redirect services, which reroute requests that cannot be filled to another host, as well as directory services and network management services. The protocols of the application layer work mainly according to the client/server communication principle. An active client contacts a passive, waiting server and transmits a service request to it. The server accepts the request of the client, processes it and sends back an answer to the client, i.e., in a positive case the requested service. Communication between the client and server therefore does not proceed symmetrically.

A few of the many important protocols of the application layer located in the TCP/IP protocol family are:

- **TELNET (TELEtype NETwork)**

TELNET (RFC 854) allows the setup of an interactive, bidirectional communication connection to a remote computer, additionally providing a command-line interface. This enables a virtual terminal to be established at the remote computer via TCP, on which commands and actions can be initiated.

- **FTP (File Transfer Protocol)**

FTP (RFC 959) facilitates the transmission and manipulation of data between two computers connected over a TCP/IP network. FTP functions according to the client/server paradigm. A client initiates the connection and requests a service. The server takes the connection request and answers the service request. The actual data transfer of the control command transmission proceeds over two different TCP ports at FTP.

- **SMTP (Simple Mail Transfer Protocol)**

SMTP (RFC 821) is a simple, structured protocol for the transmission of electronic mail on the Internet. Today, as a rule, ESMTP (Extended SMTP, RFC 5321) is used as it allows a transparent transmission of messages in different formats. SMTP is used by the Message Handling Systems (MHS) of the email service for sending and receiving messages. End systems, i.e., systems on which the end user works, use SMTP exclusively for sending email messages forwarded from a mail server.

- **HTTP (Hypertext Transport Protocol)**

The HTTP protocol (RFC 2616 among others) is used for data transmission in the World Wide Web. Just as many other protocols of the application layer, it works according to the client/server paradigm and is based on the reliable transport protocol TCP.

- **RPC (Remote Procedure Call)**

The RPC protocol (RFC 1057 and RFC 5531) is used for inter-process communication. That means, it allows a computer program to call an external, subroutine located in another addressing area. This is carried out externally and only the result transmitted to the requesting computer where it is further processed.

- **DNS (Domain Name System)**

The DNS service establishes a name and directory service that delivers the assignment between readable end system names (strings) to IP addresses for all participating systems on the Internet. The name space for end systems administered over DNS is hierarchically organized and works with local intermediate storage and proxies to ensure an efficient implementation. DNS is standardized in RFC 1123 and in numerous other RFCs.

- **SNMP (Simple Network Management Protocol)**

With the help of the SNMP protocol, network management systems can monitor, administer and control individual systems connected to the network. SNMP is standardized in RFC 3411 and in many other RFCs.

- **RTP (Real-time Transport Protocol)**

With the help of the RTP protocol (RFC 1889) real-time audio and video data can be transmitted over the Internet. For this purpose, the RTP protocol defines a separate data format for the efficient transport of a media data stream. Normally, the transport and the service quality achieved is monitored with the help of the RTCP (RTP Control Protocol). Although the protocol standard intends the TCP protocol for actual data transport, in practical application most of the time the unreliable but faster, UDP protocol is implemented. This is done to avoid inherent waiting times during connection management and error correction.

2.4 Glossary

Authentication: Serves to give proof of a user's identity. Certificates of a trustworthy authority are used to check identity in carrying out authentication. To verify the integrity of a message, digital signatures are created and transmitted with it.

Broadcast: A broadcast transmission is a transmission conducted simultaneously from one point to all participants. Classical broadcast applications are radio and television.

Circuit switching: Method of information exchange via a network. At the beginning of the information exchange an exclusive connection between the communicating terminals is established and remains in effect for the duration of the communication. Analog telephone networks, for example, work according to this principle.

Client: Designates a program that contacts a server and requests information from it. The browser employed in the WWW is in this sense a client. But there are also other clients in the WWW that contact WWW servers and download information from it, e.g., search engines or agents.

Client/Server-Architecture: An application is carried out as a collaborative effort over a network of connected, multiple computers. The server provides specific services, the client conversely requests services. Except for the actions of placing an order and responding to it, the components are independent of each other. Interfaces and the way of communicating in placing orders and responding are explicitly defined.

Communication protocol: A communication protocol (also simply protocol) is a collection of rules and regulations that determines the data format of the information to be transferred as well as the mechanisms and procedures concerning their transmission. Protocols contain agreements about the establishment and termination of a connection between the communication partners and methods of data transmission.

Computer network: A computer network (**network**) offers autonomic computing systems, which each have their own storage, periphery and computational ability, the infrastructure for data exchange. All of the subscribers are linked with each other via the computer network, therefore each has the possibility to get into contact with every other network participant.

Connection-oriented/connectionless service: A basic distinction is made between **connection-oriented** and **connectionless** services on the Internet. Before the start of the actual data transmission, connection-oriented services must establish a connection over predetermined switching locations in the network. This specified connection route is used for the duration of the entire communication. Connectionless services do not choose a fixed connection route in advance. The transmitted data packets are each, independent of one another, transmitted in potentially different ways via the Internet.

Cryptography: A branch of information technology and mathematics that is involved with the construction and evaluation of encryption. The objective of cryptography is to prevent unauthorized third parties from gaining access to confidential information.

Diffusion network (broadcast network): In a diffusion network the signal of a transmitter is received by all the computers connected in the network, with the respective time delay taken into account. Every receiver must itself determine if the message is intended for it and whether or to process it.

Flow control: Method to ensure an even and as continuous as possible data transmission between network terminals that do not work synchronously. Flow control intercedes to regulate the transmission sequence of network terminals and to slow down transmission power. This takes effect when congestion situations occur along the path to the receiver in order to avoid potential data loss.

Fragmentation/defragmentation: Because of technical restrictions, the length of the data packet sending a communication protocol in a packet-sending network is always limited below the application layer. If the message to be sent is larger than the respectively prescribed data packet length, then the message is broken down into single submessages (fragments) corresponding to the required length restrictions. To enable the original message to be put back together again correctly at the receiver after transmission (defragmentation), the fragments are provided with **sequence numbers**. This is necessary as the transmission sequence in the Internet cannot always be guaranteed.

Internetworking: The bridging of multiple, different networks that are separated from each other (LANs, WANs) into one internet. The appropriate switching computers (routers) are needed to do this. They mediate the path of a data packet through the network and ensure a secure delivery. The network appears as a homogeneous, virtual network (internet) to the user.

Internet standard: Because there were many companies and organizations involved in the development of the Internet, it was necessary to create unified protocols and interfaces to simplify the development effort. These took the form of Internet standards and were ratified in a public standardization process. Every user was principally allowed to make suggestion for future standards (Request for Comment, RFC), and thereby to steer the course of the Internet.

Internet Protocol (IP): Protocol on the network layer of the TCP/IP reference model, more precisely called **IPv4**. As one of the pillars of the Internet, IP ensures that the global Internet, consisting of many heterogeneous, individual networks appears as a unified homogeneous network. A standardized addressing scheme (**IP addresses**) enables word-wide unambiguous computer identification. IP additionally provides a **connectionless, packet switching datagram service** that cannot fulfill quality of service guarantees, but always works according to the **best effort** principle. For the communication of control information and error notification, the **ICMP** protocol is an integral component of the IP.

ISO/OSI Reference Model: A specification of the ISO that was designed and made public as the basis for the development of communication standards. It is an international reference model for data transmission consisting of seven layers. The ISO/OSI reference model has the goal of enabling different computer and protocol worlds to communicate with each other. In contrast to the TCP/IP reference model, the protocol standard underlying the Internet, the ISO/OSI reference model has become increasingly less important.

Layer model: Complex problems allow themselves be broken down hierarchically into sub-problems, built one on top of the other. The resulting layering of individual subproblems makes the modeling of the problem as a whole easier. The abstraction level increases on each individual layer. Therefore, a layer located higher in the layer model is from detail

problems handled on a lower layer. Layer models play an important role in communication technology, but also in other areas of information technology. A further representation may be seen in the corresponding **shell model**. Instead of hierarchical layers its structure is composed of individual shells.

Local Area Network (LAN): A spatially limited computer network that can only accommodate a limited number of terminals (computers). A LAN enables an efficient and equal communication for all the connected end systems. As a rule, the connected computers share a common transmission medium.

Multicasting: One source transmits simultaneously to a group of receivers in a multicast transmission. This is a 1:n-communication. Multicast is often used for the transmission of multimedia data.

Network application: An application program whose process includes the access to resources that are not available locally on the exporting computer, but rather on a remote computer across the network.

Overload (congestion): With its means of operation (transmission media, router, and other intermediate systems) a network is able to manage a specific load (communication, data transmission). If the load created in the network nears 100% of the available capacity, an overload (congestion) occurs. The network must react in an appropriate way to avoid data loss and the breakdown of communication.

Packet header: In a packet switching network, the communication protocols implemented require the fragmentation of the information to be transmitted into individual data packets. In order to ensure that the data packets reach the designated receiver in the correct form, and can be reassembled into their original information, command and control information is added to the data packet in a so-called data packet header.

Packet switching The primary communication method in digital networks. The message is broken down into individual data packets of a fixed sized. The packets are then sent individually and independently of each other from the transmitter, over any existing switching centers, to the receiver. A distinction is made between **connection-oriented** and **connectionless** (datagram network) packet switching networks. In connection-oriented packet switching networks, a connection is established over a fixed packet switching center established in the network before the start of the actual data transmission. Conversely, in connectionless networks there is no predetermined connection path.

Protocol stack: The various subproblems of network communication are each handled by special protocols. These must all work together smoothly to solve the problem of network communication as a whole. In order to guarantee the functioning of this interplay, the development of network protocol software is seen as a comprehensive task. To solve it, an accompanying **family of protocols** (protocol suite) was developed that addresses each subtask and integrates them efficiently with each other. The entire problem of network communication may be represented with the help of a **layer model**. As the individual protocols of a protocol family are each assigned to a specific layer, the term **protocol stack** is used. The most well-known protocol stacks are the TCP/IP protocol suites of the Internet and the ISO/OSI layer model, which often serves as an instructional example.

Quality of service: Quantifies the performance of a service offered by a communication system. It is described by means of the performance of service, quality attributes, performance fluctuation, reliability and security, which in each case are specified via individual, quantifiable service quality parameters.

Reference model: An abstract model that serves as the basis for deriving more specialized models or concrete implementations. Reference models are often used as general objects of comparison with other models describing the same technical concept. In the area of computer networks there exist two well-known reference models. The ISO/OSI reference

model, which is primarily used today for didactic purposes, and the model actually implemented in the Internet: the TCP/IP reference model.

Request for Comments (RFC): New technologies pertaining to the Internet under discussion by experts are recorded in so-called RFCs. In the course of the Internet standardization process there evolved a collection of consecutively numbered documents where technologies, standards and miscellaneous information connected to the Internet were documented and standardized.

Router: A switching computer that is capable of connecting two or more subnets with each other. Routers work in the transport layer (IP layer) of the network and are able to forward arriving data packets along the shortest route through the network based on their destination address.

Routing: Along the path of a WAN there are often multiple switching elements between transmitter and receiver. These carry out mediation of the transmitted data to the respective receiver. The determination of the correct path from transmitter to receiver is called routing. The dedicated switching centers (**routers**) receive a transmitted data packet, evaluate its address information and forward it correspondingly to the designated receiver.

Server: Describes a process that clients request in order to receive information or be provided with resources. The computer on which a server process runs is often known as the server.

Security: In network technology the term security encompasses different security objectives (quality of service parameters), describing the degree of integrity and authenticity of the transmitted data. Among the most important goals of security are: **confidentiality** (no unauthorized third party being able to eavesdrop on data communication between the transmitter and the receiver), **integrity** (ensuring accuracy of the received data), **authentication** (guarantee of the identity of the communication partner), **liability** (legally binding proof of a completed communication) and **availability** (guarantee that an offer of service is in fact available).

Transmission Control Protocol (TCP): Protocol standard on the transport layer of the TCP/IP Reference Model. TCP provides a reliable, connection-oriented transport service upon which many Internet applications are based.

TCP/IP Reference Model (also TCP/IP protocol suite, TCP/IP communications model): Designates a communication layer model for the Internet. The TCP/IP reference model is divided into 5 protocol layers and enables different computers and protocol worlds to communicate with each other via standardized interfaces on the Internet.

Topology: The topology of a computer network is understood as the geometric form of the distribution of individual computer nodes in the network. Widespread topologies for computer networks are the **bus topology**, **ring topology** and **star topology**.

Wide Area Network (WAN): A freely scalable computer network that is not limited by spatial or capacity restrictions. Individual subnets are connected with each other by switching systems (**routers**), which coordinate data transfer in the WAN. The WAN technology supplies the foundation for **internetworking**.

Internetworking

Technological Foundations and Applications

Meinel, C.; Sack, H.

2013, XIII, 903 p. 426 illus., Hardcover

ISBN: 978-3-642-35391-8