2 OSI Reference Model

arrangement

For the realization of computer networks, a variety of tasks needs to be solved. These tasks will be explained below in light of several issues.

One issue is how to transmit bits via media. For example, we can distinguish different voltage values when using copper cables or we can use electromagnetic waves for transmission over the air. The precise use of the medium, particularly how a logical “1” and a logical “0” are represented, has to be defined.

Another problem to be solved is routing. For example, there are a number of possible routes to transfer data units from Germany to Australia. You can imagine that the data is routed via Dubai and Singapore or via North America and the Pacific. It is necessary to clarify how these routing decisions are made and which criteria play a role in these decisions.

There are also many applications such as WWW, e-mail, VoIP, etc. that are based on the Internet. It has to be determined here how they can access the network and how they have to be programmed.

Given these selected issues, you can see that there is a set of tasks that have to be handled. In practice for computer networks, several layered models have become prevalent for this purpose. With these layered models you can assign different tasks to different layers and then try to solve the tasks separately from one another.

The OSI (Open Systems Interconnection) reference model has been developed by ISO for the modeling of communication. It will be presented in the following and will serve for subsequent structuring of the course. The OSI model consists of seven layers that build on top of each other.

In the online version an video is shown here.

Link to video : http://www.youtube.com/embed/rPNnJaqoLBw
OSI Reference Model

2.1 Rationale

In the abbreviation OSI, the O stands for open, i.e. it is a freely accessible standard. In the beginning of the OSI model development in 1977, the situation was completely different from today. This situation is described briefly here.

At that time the era of mainframe computers that filled entire rooms had already passed, but so-called minicomputers were still devices that had the external dimensions of a large refrigerator. The focus was on hardware at the time, and manufacturer-specific operating software was delivered with the computer. The computers already offered networking possibilities for long-distance communication over telephone networks, which however was limited to the respective manufacturers. If, for example, a data center used an IBM computer, it could be networked with other IBM computers in other data centers. But if other devices from Digital Equipment or Siemens, for example, were used, they could not be networked with IBM devices this way. This meant you were locked in the world of your vendor.

The aim was therefore to create a generally accepted way of establishing communication so that devices from any manufacturer could communicate with each other. Similar to how current international meetings happen in English today, a kind of common language was needed.

annotation

At the time, the American National Standards Institute (ANSI) got the task from ISO to develop a proposal for an appropriate model. ANSI then invited companies to provide submissions for such a model, but only Honeywell actually offered a model. This was an enhancement of IBM's Systems Network Architecture. This model was then later refined to become the OSI model. This brief outline (see page 20 in *Doyl 2006*) is only

to show that the model is not a “holy grail.” Coincidences also played a role in how the model looks like.

2.2 Basic Principles

arrangement

The OSI model was established in 1983 by the ITU and in 1984 by the ISO as a layered model. This means that a lower layer provides basic functionality upon which a higher layer is built and additional functionality is added. This layer is then available to an even higher layer.

The definition of the layers is based on the following considerations.

- It is desirable to be able to treat all the tasks to be solved separately from each other in order to get complexity under control. Therefore closely related tasks are grouped in one layer, and tasks that differ from each other are put in different layers.
- Layers can be accessed via interfaces, which will be described shortly. Within a layer, you can make changes (e.g., use improved algorithms) without the need to change other layers, as long as the interface remains unchanged.
- All layers are needed for end-to-end communication of applications according to the model. A layer can be divided into sublayers, and sublayers can be omitted if they are not necessary.

Two concrete examples are provided here to illustrate the advantages of the second point.

- The first WLAN standard from 1997 specified a bit rate of 1 Mbit/s for wireless transmission. The bit rate was significantly increased over time so that the standard IEEE 802.11ac from 2014 stipulated bit rates over 1 Gbit/s. The changes to increase bit rates only affected the lowest layer (Physical Layer) so all other layers could remain unchanged.

- At present a transition is occurring from Internet Protocol version 4 to version 6. This only affects the third layer of the model (Network Layer) so that the layers above and below can theoretically remain unchanged. The qualification “theoretically” is added to indicate that in real implementations there is not such a strict separation between the layers as in the model. Such a strict separation would lead to inefficient implementations where, for example, copying operations would often be necessary in memory.

2.2.1 Structure of the OSI Model

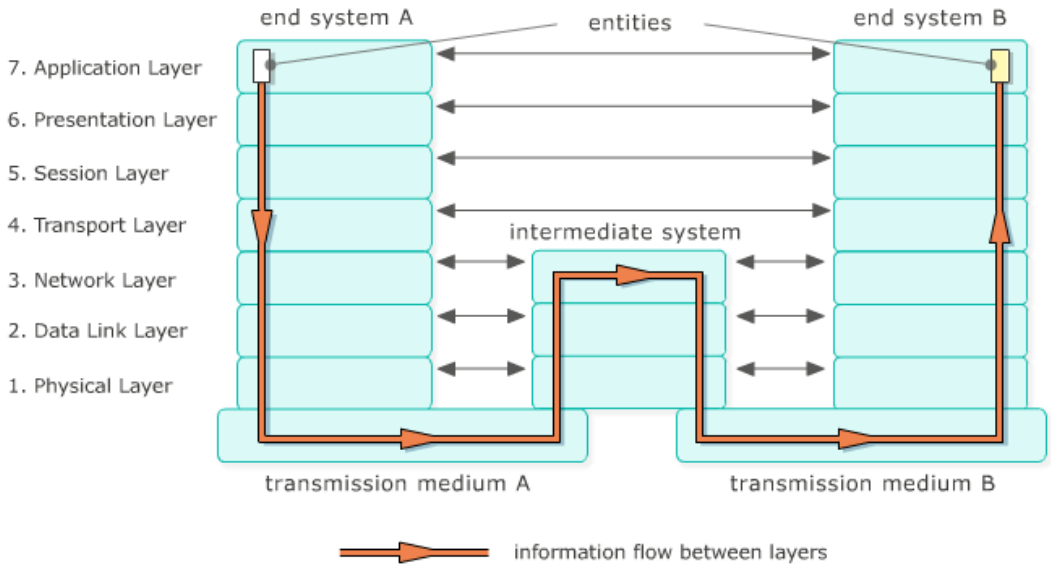
As already mentioned, the OSI model consists of seven layers. The functions of the respective layers are performed by **instances** (also called entities), which can communicate both vertically with the instances above and below as well as horizontally with instances at the same layer in a spatially separated system. A communication network that consists of end systems and intermediate systems results in the ISO/OSI reference model structure depicted in the following illustration.


In the online version an animation is shown here.

The OSI reference model

Begin printversion

Each layer of the reference model is represented by an entity. The entities implement their assigned functionality, e.g. by running a software program and by communicating with a neighboring system.



Each entity of a layer N needs the services of layer N-1 to be able to implement its way of communication. Layer N-1 itself accesses layer N-2 and so on. The process continues until Layer 1 is reached. The real transmission happens on this layer using the transmission medium. For traversing the layers in the described manner, the entities of the layers communicate with their neighboring layer entities in the same system - i.e. from top to bottom and from bottom to top, respectively – via so called service primitives.

To provide a service according to a specific layer, the entity of this layer communicates with the entity in the partner system (here for example in end system B) on the same layer. This is done by the exchange of layer-specific protocol data units.
End printversion

End systems as well as an intermediate system (transit system) can be seen in the video above. An intermediate system implements only a subset of the layers; there are different types of intermediate systems that are presented in the section on intermediate systems. In the illustrated case, the intermediate system implements the first three layers (in practice this applies to so-called routers).

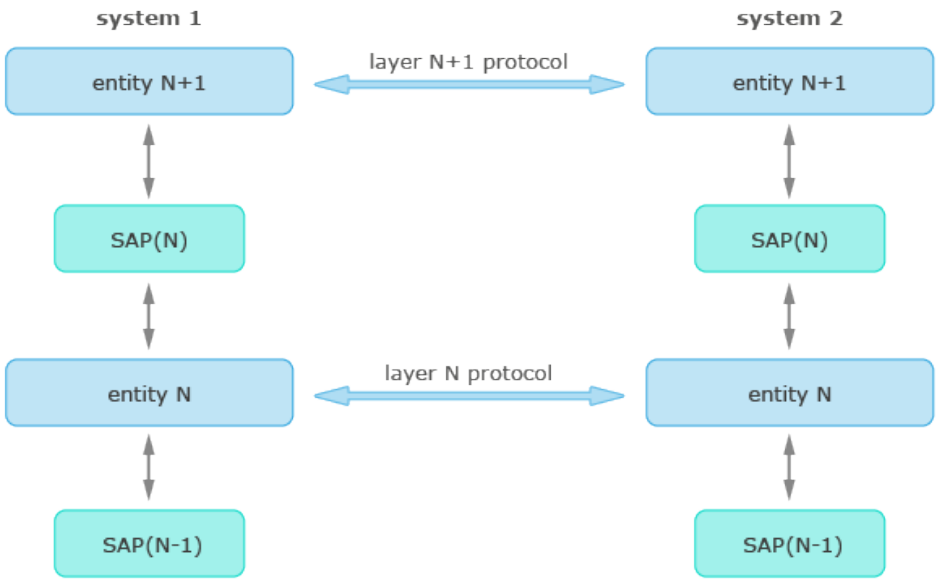
2.2.2 Services

In each layer of the OSI model, there are instances (entities) which realize layer-specific **services**. A service is a functionality that is offered to a higher layer. According to ISO an instance is an "active element", e.g., an executable computer program, which can work together with other instances. Two vertically adjacent instances are also called upper

and lower instances; two instances at the same layer in remote systems are called peer instances (peer entities).

The arrangement (see figure below) is a strict hierarchy, i.e., an instance at layer N can only use the service of an instance in the layer N-1 directly below, and it can only offer its services to an instance at the layer above (layer N+1). This principle is, however, unfortunately often violated in practice. The services provided by a layer are made available at a **service access point (SAP)**. The SAP should thus be understood as a logical interface between two layers. In order to ensure that the proper instances are addressed in communication, unique addresses have to be used.

The communication between instances occurs in both vertical and horizontal directions. The vertical communication is based on the exchange of **service primitives** between the instances in the same system; the horizontal communication is based on the standardized, layer-specific **protocols**.



Interaction with adjacent layers

2.2.3 Protocols

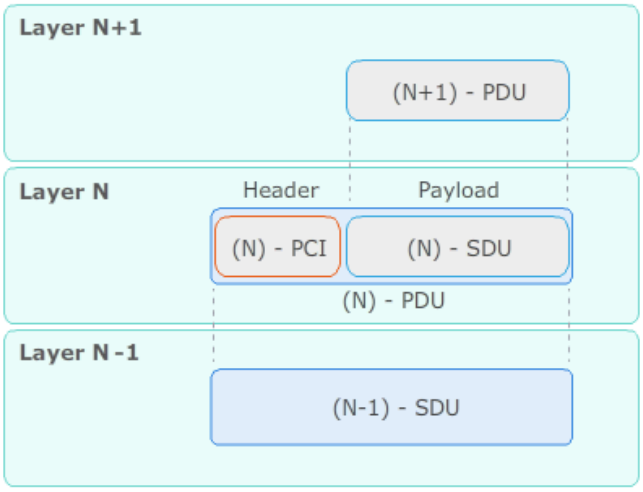
A protocol consists of a definition of formats of the possible **protocol data units (PDU)** as well as the order of the communication sequence. To implement the protocol each instance logically communicates with its adjacent peer entity in a remote system through the exchange of protocol data units.

A PDU generally consists of three areas: header, payload and trailer. The trailer may be omitted.

In the online version an animation is shown here.

Protocol data units in the OSI reference model

Begin printversion



Audiotext

The figure shows three layers of the OSI model: Layer N and its neighboring layers. From the higher layer N+1 a protocol data unit is provided to layer N. Once this PDU enters into layer N, it is called a service data unit because it wants to make use of the services on layer N. The whole SDU in then the payload of layer N. Now protocol control information (PCI), which is also called the header, is added to the SDU. PCI and SDU of layer N are combined and form the protocol data unit of layer N. This PDU is now sent to the underlying layer N-1 to use its services.

End printversion

A protocol data unit from layer N+1 is considered in the lower layer N as a data unit that wants to use the services of layer N. Therefore the (N+1)-PDU is referred to as a layer N **service data unit (SDU)**. Layer N adds its specific **protocol control information (PCI)** to the (N)-SDU and provides these data to the lower layer N-1 as a new N-PDU. Finally the data with all protocol control information for all participating layers are transmitted to the receiver via the transmission medium.

At the receiver, the reverse process takes place: the lowest layer processes the protocol control information relevant to it and carries out the necessary functions. Once it has finished its processing, it removes the protocol control information and transmits the rest of the received data to the protocol instance of the next higher layer. After all layers have been run through, the target instance receives the data from its peer entity.

The following section shows a practical example of this. Although the individual names used for the layer protocols anticipate later chapters, it already becomes very clear here how the data sets to be transmitted become increasingly large due to repeated embedding in the protocol structure at each layer.

This process of adding or removing of header and trailer information is referred to as "**encapsulation**" (transmitting end) or as "**decapsulation**" (receiving end).

2.2.4 Encapsulation and Decapsulation

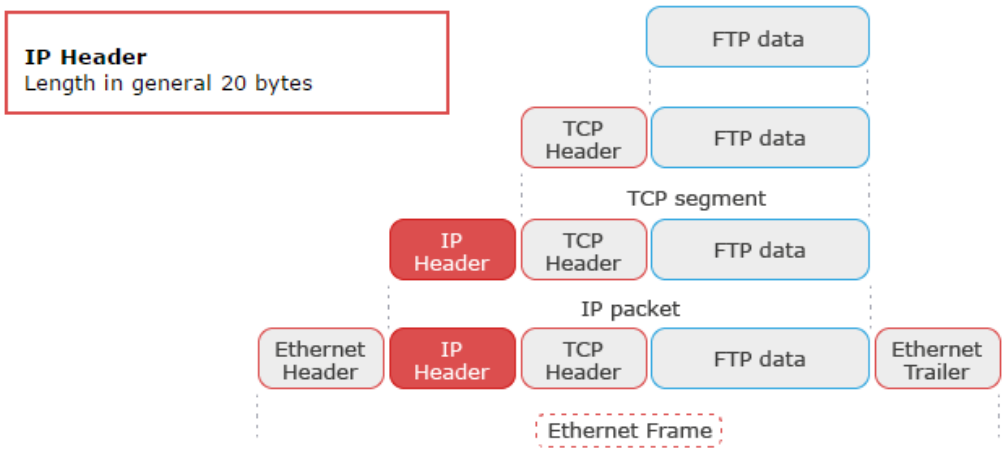
The figures show examples of encapsulation and decapsulation for transmission of an FTP data unit. FTP is a protocol at layer 7 for the transmission of files between computers. In the example, on layer 2 of the model an Ethernet network is used, which is typical for local fixed networks.

The following figure illustrates encapsulation on the transmitting end.

In the online version an rollover element is shown here.

Encapsulation process

Begin printversion



TCP header: Length in general 20 bytes
IP header: Length in general 20 bytes
Ethernet header: Length = 14 bytes
Ethernet frame: Length between 64 and 1518 bytes
Ethernet trailer: Length = 4 bytes
End printversion

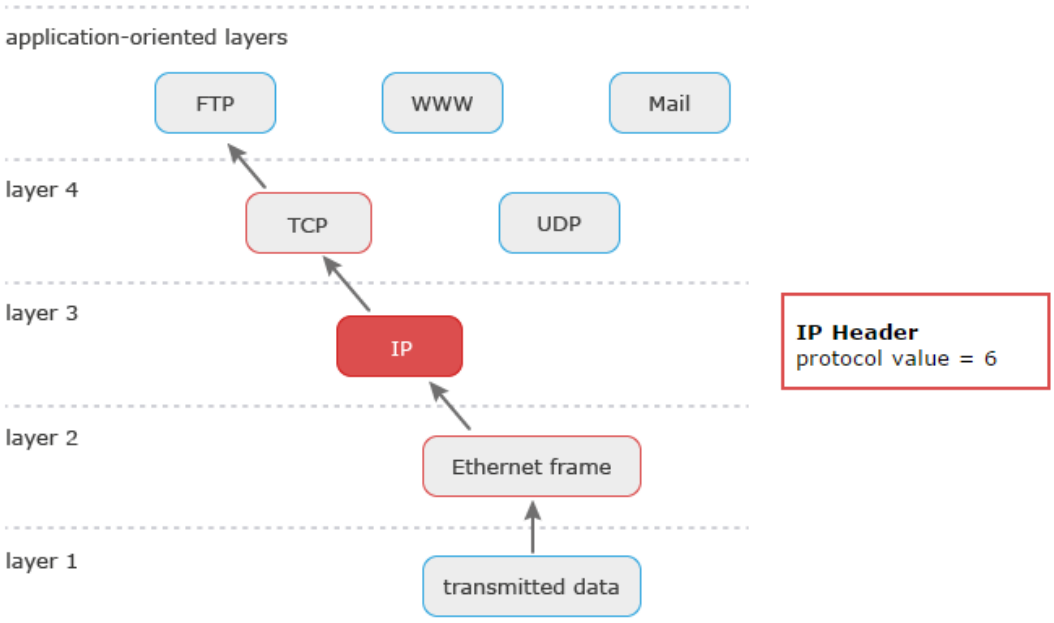
- The FTP data is passed to the **TCP** (Transmission Control Protocol). TCP adds the **TCP header** (in general **20 bytes**) to the data.
- The TCP segment is passed to the **IP** (Internet Protocol). IP adds the **IP header** (in general **20 bytes**).
- The **IP datagram** is passed to Layer 2 (**Ethernet**). There the Ethernet header (14 bytes) and the Ethernet trailer (4 bytes) are added - i.e., in sum **18 bytes** are added.
- The **Ethernet frame** can now be sent. The length of the Ethernet frame must be between 64 bytes and 1518 bytes; a maximum of **1500 bytes** of payload data can therefore be transmitted in an Ethernet frame.

An Ethernet frame with FTP data is received by the Ethernet network card:

In the online version an rollover element is shown here.

Decapsulation process

Begin printversion



TCP/header: Port number = 21

IP header: Protocol value = 6
Ethernet frame: Type = 0x0800
End printversion

There must be a field in every header, which specifies the protocol of the layer above that should be addressed:

- If in the Ethernet frame 0x0800 is specified as the **type**, the frame will be forwarded to IP.
- If in the IP header the value 6 is specified as **protocol**, the IP datagram will be forwarded to TCP.
- If in the TCP header 21 is specified as the **port number**, the FTP server will be addressed.


annotation

As we have seen in the example, data units are named differently for each protocol: frame (for Ethernet), packet (IP), datagram (IP, UDP), segment (TCP) and message (for applications). The general term is data unit. Precisely speaking, the data is called a protocol data unit (PDU).


notice

You can see the complexity of the Internet in the following video provided by the "Warriors Of The Net" project.

The video provides an amusing description of how communication takes place on the Internet. How do data streams on the Internet reach their destination? What are packets or firewalls? How is/are network(s) constructed?

The video lasts about 13 min. It is available in different resolutions! We recommend you watch it at least once.

We would like to take the opportunity here to thank Warriors of the Net again for our successful collaboration!


In the online version an video is shown here.

Link to video : http://www.youtube.com/embed/PBWhzz_Gn10
Animated video Warriors Of The Net

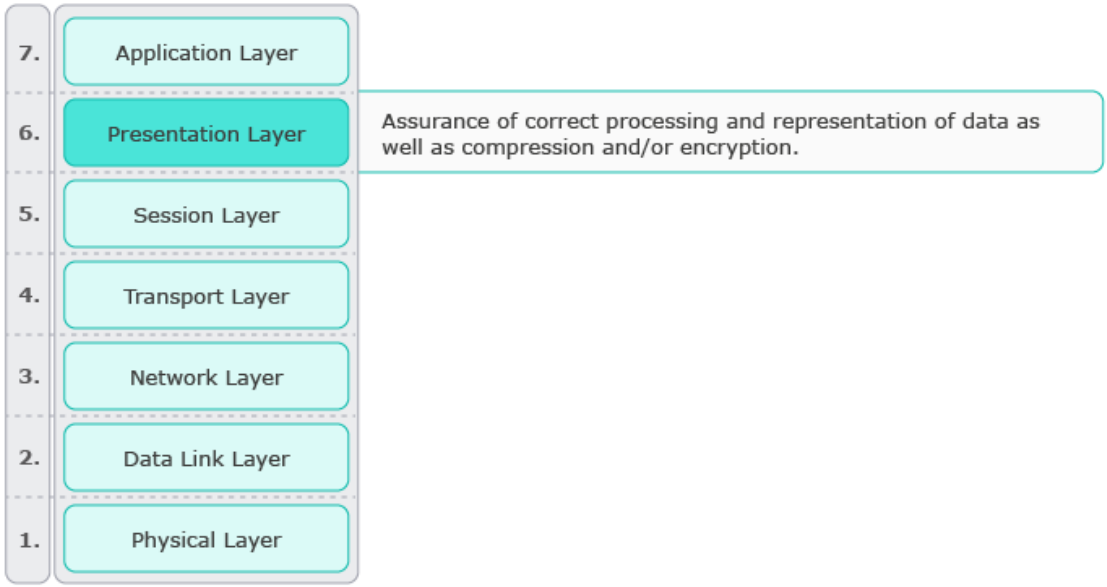
2.3 The Layers of the OSI Model

The following figure provides an overview of the seven layers of the OSI model.

In the online version an rollover element is shown here.

The seven layers of the OSI reference model

Begin printversion



7. Application layer: Implementation of application processes.
6. Presentation layer: Assurance of correct processing and representation of data as well as compression and/or encryption.
5. Session layer: Implementation of communication sessions.
4. Transport layer: Transfer of data units between one process on one end system to the corresponding process on the other end system.
3. Network layer: Transfer of data units between end systems over the network.
2. Data link layer: Transfer of data frames between neighboring systems. Detection and correction of transmission errors.
1. Physical layer: Specification of physical characteristics, bit transmission between neighboring systems.
End printversion

2.4 Transport-Oriented Layers

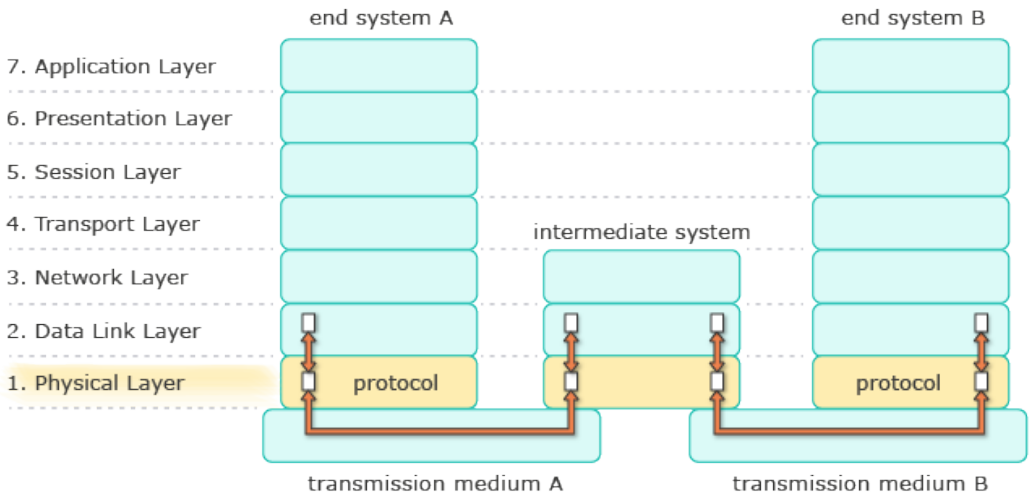
arrangement

The first four layers of the OSI model, which are referred to as transport-oriented layers, enable communication between applications on different end systems. This communication is independent from the meaning of the transmitted data.

Layers 1-4 are implemented in the **kernel** of current operating systems. That is why a computer is already recognizable in a network when the operating system has started.

The descriptions explain the respective issues to be solved and the services offered. Not every implementation of a layer has to have every feature. This applies in particular to the transport layer where you can choose between the protocols TCP and UDP. UDP realizes only a portion of the tasks of the layer, while TCP has significantly more features.

2.4.1 Layer 1 - Physical Layer



The Physical Layer

The lowest layer of the reference model is generally responsible for the conversion of a bit stream to signals that can be transmitted via a medium. The bits are converted to signals on the transmitting end and then reconstructed as bit stream on the receiving end. A medium can for example be a copper cable (twisted-pair cable, coaxial cable), a fiber-optic cable or the air. This means a property of the medium has to be found which can be used for the transmission of bits. For example, if you know that you can distinguish between different voltage values in a copper cable, you can allocate different logical states to these voltages. You could then define 0 V as logical 0 and 5 V as logical 1. This definition is, however, arbitrary and could also be done in the opposite way.

The transmitter and receiver have internal clocks that work independent from each other, so a mechanism for synchronizing the clocks is needed (additional transmission of the time signals is often too expensive or difficult to realize). You can, for example, transmit a known sequence of bits that always form the beginning of a transmission. The receiver can use these bits to adjust its internal clock to the transmitter's internal clock. This is necessary for the reliable recognition of the following unknown bits.

Transmission in this layer is specified in a way so that the bits are usually reconstructed properly. However, this is not guaranteed and nothing further is done in this layer to ensure proper reconstruction. There are no buffers on the transmitting or receiving end, where a buffer on the transmitting end could temporarily store the data for possible retransmission.

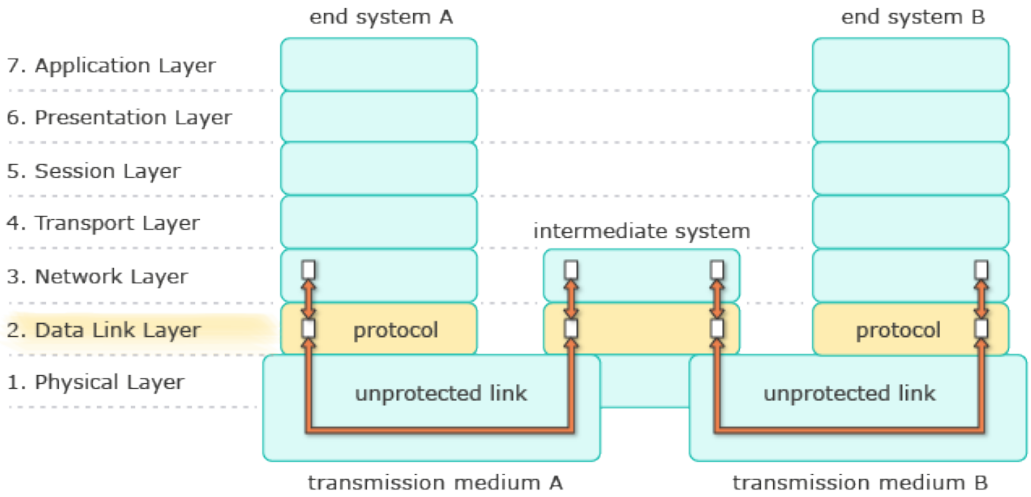
Service:

- Transmission of unstructured bit sequences via a medium such as a copper cable, fiber-optic cable or air.

Protocol:

- Definition of properties for the medium used (for example, which attenuation characteristics a cable type may have); for wireless communication, the air itself can, of course, not be influenced
- Definition of plug standards
- Representation of the values 0 and 1 (for example, as voltage values or phase shifts)
- Synchronization between transmitter and receiver (reliable recognition of the start and end of a bit)

2.4.2 Layer 2 - Data Link Layer



The Data Link Layer

The main tasks of this layer are in general safeguarding against transmission errors and medium access control. By safeguarding against transmission errors we mean that mechanisms are provided at this layer that compensate for transmission errors. From the perspective of the higher layers this has the effect that data is never lost in transmission over a link. For this purpose, it is necessary to no longer consider the bits to be transmitted as unstructured bit stream but to group the bits into data units. These are called frames. During the transmission of frames, some bits can arrive incorrectly or entire frames can be lost. By using checksums, i.e., additional control bits, which are attached to the frame, the receiving end can determine whether the frame has arrived correctly. If the frame has been received incorrectly and in a case where a frame does not arrive at all, the receiving end will not send a confirmation of the frame to the transmitter. The transmitter, which awaits confirmation for all frames, would then resend the frame. A prerequisite for resending frames is that frames that have been sent but are not yet confirmed have to be temporarily stored by the transmitter in a buffer. So the transmitter has the possibility to resent them. In addition to error correction through retransmission, forward error correction can be used. This means the additional control bits sent in the frame do not only allow for bit error detection, but also directly allow for their correction on the receiving end.

On the Data Link Layer, there is also flow control, which means that the receiver can send a message to the transmitter that the data should not be transmitted in such a dense sequence. This may be necessary if the transmitter is significantly more powerful than the receiver.

The second main task of this layer is medium access control (MAC). This involves finding possibilities for multiple participants to use a shared medium. In this regard, there are time, frequency, code and space division multiplexing. Combinations of these methods are possible and are frequently used. Frequency division multiplexing means, for example, that the participants send data on different frequencies.

To distinguish between participants, addressing is introduced, where each participant gets a unique identifier. Apart from uniqueness, the addresses on this layer can be assigned in an arbitrary manner. For WLAN, Bluetooth and Ethernet, so-called **MAC addresses** are used. These are permanently assigned by the network card manufacturer during the production process.

Service:

- Reliable transmission of a structured bit sequence (frame) via a “safe channel”

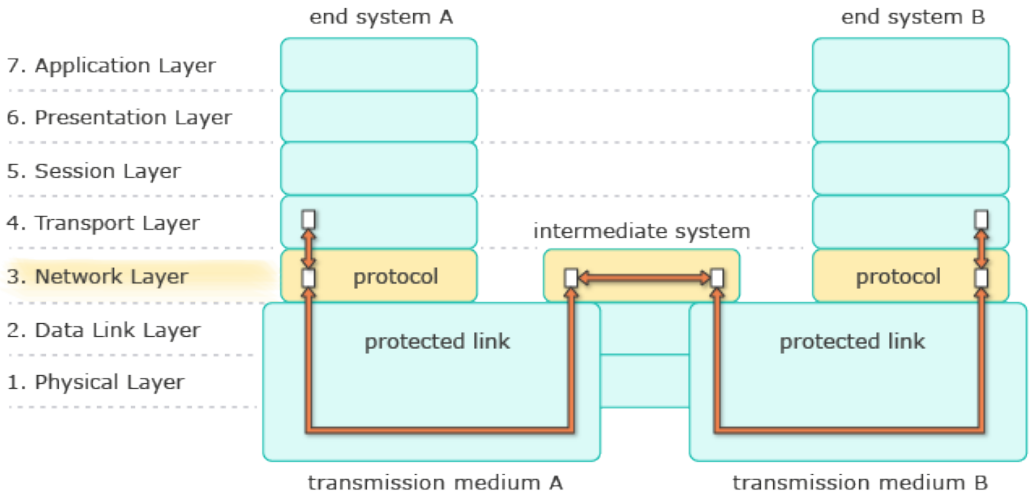
Protocol:

- Medium access control, i.e., protocol mechanisms for common use of a shared medium by different participants.
- Transmission of bit sequences in frames
- Addressing (for direct communication; addresses must be unique but otherwise do not have to be structured in a particular way)
- Detection and correction of transmission errors
- Flow control for handling overload situations on the receiving end
- Buffering of data on the transmitting and receiving ends
- Implemented primarily in hardware on adapter cards

annotation

Not all properties of this layer actually need to be implemented in real systems. It is possible in particular to sometimes omit bit error correction. This makes sense for example in broadcasting systems (DAB, DVB) where the receiver is not able to provide feedback. In this case, only forward error correction is carried out without retransmitting data.

2.4.3 Layer 3 - Network Layer



The Network Layer

The third protocol layer is mainly responsible for the routing of data and their forwarding in communication networks. It should allow any devices to communicate with each other even if they are not directly connected via a transmission link. The addresses from the Data Link Layer cannot be used for routing on this layer because they do not contain location information. Therefore different addresses are introduced, which are structured hierarchically. In data networks today these are nearly everywhere the addresses belonging to the Internet Protocol, i.e., IP addresses. Such addresses consist (in version IPv4) of 32 bits. The bits at the start are used to distinguish between networks, and the bits at the end are used to distinguish the end systems in these networks. The allocation of bits for these two purposes is variable.

Another task of this layer is fragmentation and its reversal, reassembly. It sometimes happens that data units on this layer, which are called packets, are larger than the maximum payload data length on the Data Link Layer. In this case, a packet must be transmitted in several frames.

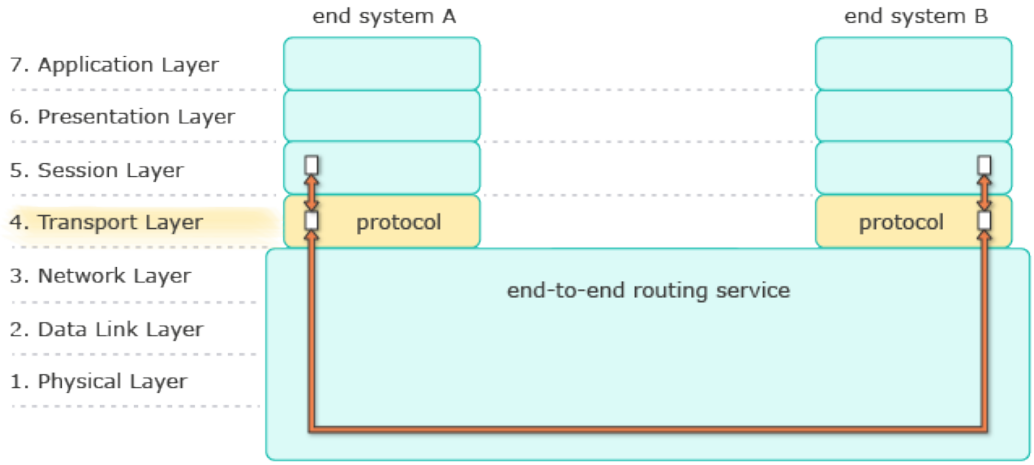
Service:

- Unreliable end-to-end communication between end systems via networks

Protocol:

- Determining a route through the network
- Multiplexing of end-system connections via links
- Addressing of end systems and intermediate systems (in hierarchical manner)
- Simple error detection, only reaction to serious error situations
- Fragmentation: split-up of a packet into frames

2.4.4 Layer 4 - Transport Layer



The Transport Layer

The fourth layer is necessary for safe end-to-end transmission so that data sent by the original transmitter actually arrives at the end receiver. Although data forwarding is safe on each individual transmission link through the protocol mechanisms of Layer 2, it can still happen that data is lost along the way.

This can happen especially if intermediate systems are overloaded. These intermediate systems receive the data units but discard them during internal processing. Such a situation can occur when too many transmitters want to send too much data so that the network is overloaded. In this case the buffers in the systems in the network fill until ultimately data can no longer be received and the arriving data is discarded. Appropriate mechanisms are introduced for congestion control, which ultimately ensure a reduction of the data transmission rate. In addition to congestion control, flow control is also provided. This mechanism does not look at the data transfer on a per-link basis (as in Layer 2); instead, it ensures that the original transmitter does not overload the final receiver.

In practice, mainly TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) are used in this layer. While TCP implements the mechanisms as described for error, congestion and flow control, UDP is much simpler and instead only contains simple error control similar to IP.

On the Transport Layer, the communication is no longer between end systems, but different applications on end systems should be able to communicate with each other. To distinguish between these applications, an additional way of addressing is required. In practice these addresses are port numbers for which there are standardized assignments to applications (defined by IANA).

Service:

- End-to-end communication channels between applications

Protocol:

- Addressing of applications (port numbers)
- Virtual connections over connectionless datagram services
- Sophisticated error detection and correction between applications
- Congestion control to prevent overloading of the network
- Flow control between applications
- Different qualities of service possible (especially by choosing between TCP and UDP)

notice

The term “Transport Layer” can lead to a misunderstanding. Transmission in the network is already implemented at the lower layers, i.e., these lower layers provide the “transport” of data. The instances of the Transport Layer can only be found in the end systems. Intermediate systems such as routers or switches do not know the Transport Layer.

2.5 Application-Oriented Layers

arrangement

The OSI Model provides three layers, which group the tasks to be solved for applications. The **Session Layer** (Layer 5) ensures communication between two sides in a connection context. This can be used, for example, in the case of a money transfer to ensure that the participant who would like to carry out the transfer was previously authenticated.

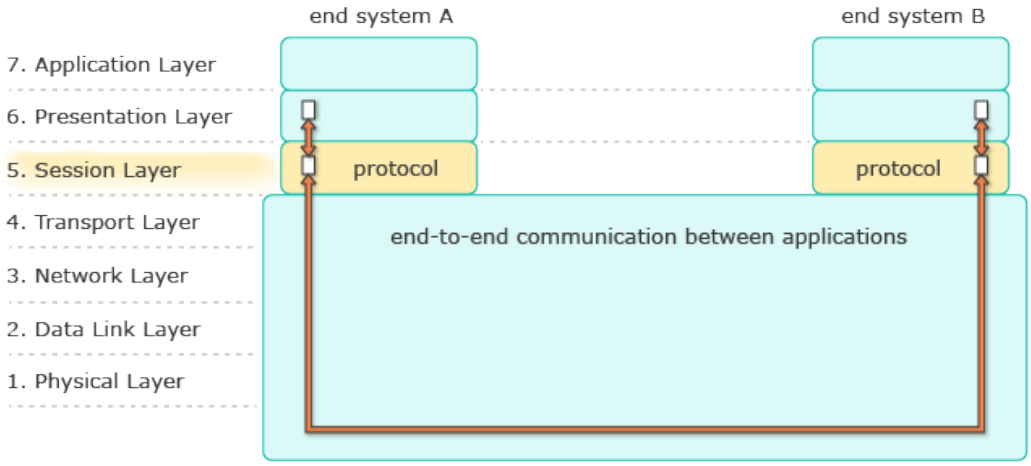
The **Presentation Layer** (Layer 6) allows the application to be adapted to different end systems and thus to abstract from their properties (such as when one processor looks at integers as 16-bit values and another as 32-bit values). This task has to be seen in the context of isolated manufacturer worlds (such as IBM, DEC, Siemens), whose interaction was supposed to be enabled by the OSI Model. The encryption of data during transit, e.g., via Transport Layer Security (TLS), and data compression can be also assigned to ISO/OSI Layer 6. The actual applications are then implemented on the **Application Layer** (Layer 7). Such a separation into these three layers has, however, not been adopted in reality.

Therefore every network-based application is based directly on the Transport Layer, so decisions have to be made again and again about the use of the Transport Layer. Through the use of program libraries (APIs), it is often not obvious that no general solution exists. For example, cookies on browsers and web servers are an example of session management that is in practice implemented at Layer 7 although it can logically be mapped to Layer 5.

2.5.1 Layer 5 - Session Layer

A communication relationship between two application processes for collaboration on a common task is called a **session**. This layer provides a general design for such sessions.

To illustrate this, we present an example of an online shop where, for example, you can buy office materials. During shopping a website visitor places various items in a virtual shopping cart, and the contents of the shopping cart can change all the time. At the end the visitor pays and thereby ends the session. It is important here that during the session the context between the individual queries is managed so that in particular the payment at the end matches to the selected items.



The Session Layer

Service:

- Provisioning of sessions between participants

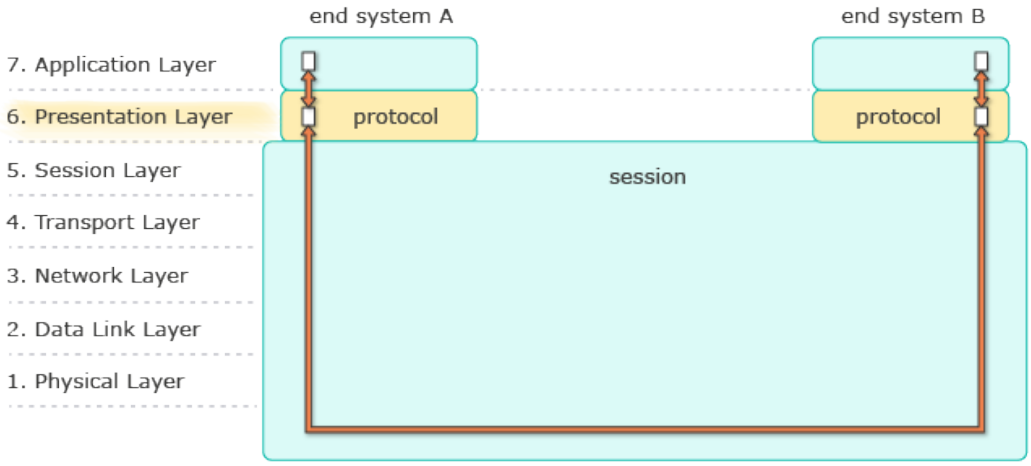
Protocol:

- Establishment and decommissioning of session contexts
- Authentication methods
- Reset agreements (so that consistent states are ensured even if connections are broken)

2.5.2 Layer 6 - Presentation Layer

The Presentation Layer allows for abstraction from different hardware. For example, you would like to be able to program an application once and then have it compiled automatically for different systems. Differences between the systems, such as whether an integer variable is 16 bit in one system and 32 bit in another, or how the bytes of the variables are stored in memory (Little Endian, Big Endian), should not play any role in the programming. While the above-mentioned problems have existed for a long time, the issue of how to deal with different screen resolutions and types of devices (smartphone, notebook, etc.) became more important in the last years.

In addition to the points mentioned, this layer is also concerned with issues of data compression and encryption for secure transmission.



The Presentation Layer

Service:

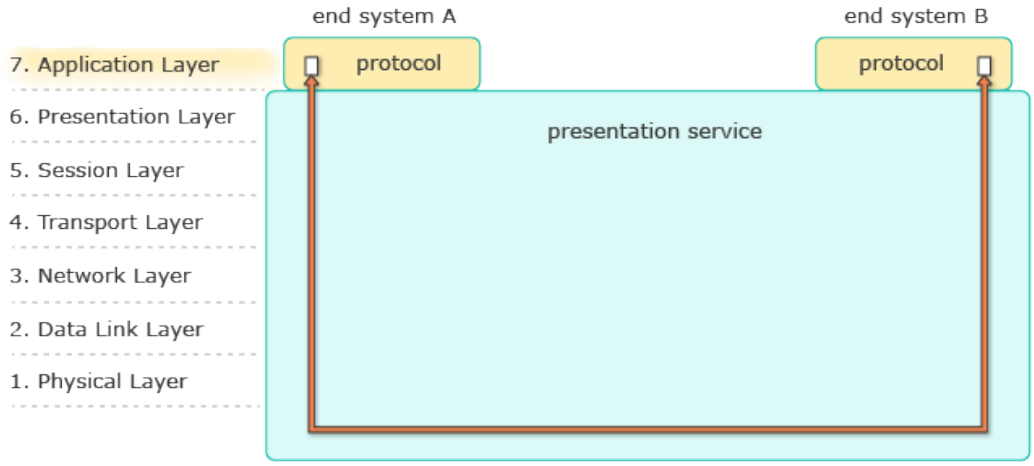
- Implementation of applications for various systems

Protocol:

- Definition of general data types
- Implementation in specific system
- Data compression
- Encryption

2.5.3 Layer 7 - Application Layer

The seventh and final layer of the OSI model is the **Application Layer**. It is the interface between people and applications. Users only see the programs that they are familiar with. This includes e-mail programs, WWW browsers, file-transfer programs and much more. The other layers below this, such as how the underlying network works, remain concealed from the user.



The Application Layer

Service:

- Provisioning of applications for human users

Protocol:

- Application specific

annotation

Applications can be built on top of each other. For example, WWW and e-mail use DNS as a basic service.

2.6 OSI Model in Practice

arrangement

In the beginning, the OSI Model was not only regarded as a theoretical model for structuring tasks, but was also supposed to be implemented with OSI protocols for the different layers. In the 1980s, however, these protocols came into competition with the well-functioning world of TCP/IP protocols. At the end of the 1980s, the TCP/

IP protocols ultimately prevailed in practice because the few available OSI protocol implementations were not performing as good as expected.

One of the few OSI protocols still relevant today is IS-IS, which is used for routing in the networks of large Internet service providers.

In the online version an video is shown here.

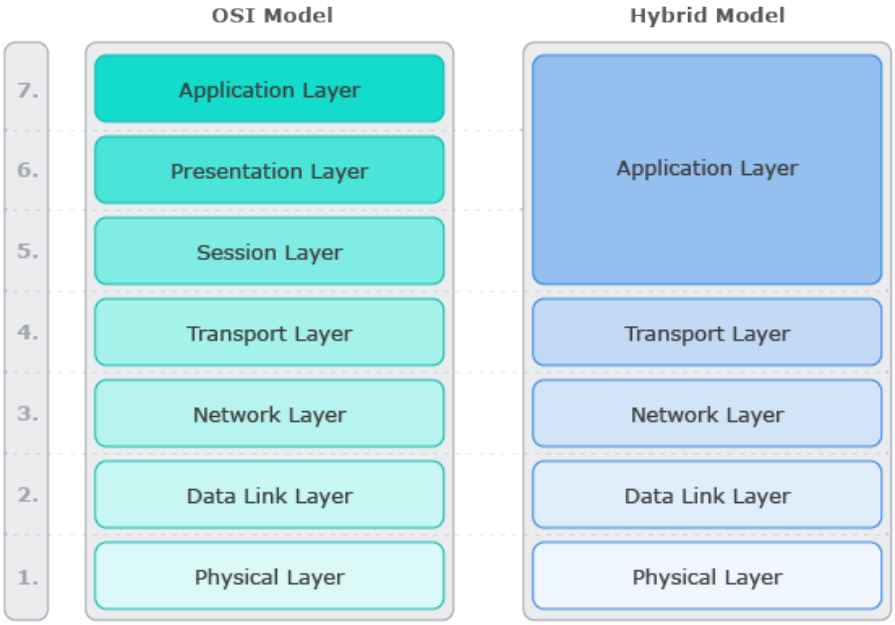
Link to video : http://www.youtube.com/embed/P6NET5Y0Q8I
OSI Model / Internet Model / Hybrid Model

2.6.1 Comparison with the Internet Model

The Internet Model was developed in context with the protocols TCP and IP. It consists of only four layers. The lowest layer of the model is a combination of the layers one and two of the OSI Model and is called the Subnetwork Layer. The second layer is called the Internet Layer. It is named after the Internet Protocol and is equivalent to the Network Layer. Then there is the Transport Layer, which corresponds to the Transport Layer in the OSI Model. The highest layer in the Internet Model combines the three upper layers of the OSI Model and is called Application Layer.

2.6.2 Hybrid Model

The Hybrid Model is a compromise that best reflects the situation in reality. It makes sense to maintain a separation between the transmission of bits in the Physical Layer (Layer 1) and the safeguarding of transmissions in the Data Link Layer (Layer 2) as provided for in the OSI Model because these tasks are clearly separate from each other. On the other hand tasks are not handled in the isolated way as specified in the OSI Layers 5-7 (Session, Presentation, Application) when programming network applications. Therefore the Hybrid Model has an Application Layer that is based directly on the Transport Layer (Layer 4).



Layers of the Hybrid Model

2.7 Intermediate Systems

In the online version an video is shown here.

Link to video : http://www.youtube.com/embed/FENV-NO9wEE
Intermediate systems

There are many different devices that function as intermediate systems, which are distinguished from one another based on the level of the OSI Model on which they operate. Basically, the following **network components** are distinguished:

- **Repeaters** or **hubs** operate on Layer 1
- **Bridges** or **switches** operate on Layer 2
- **Routers** operate on Layer 3
- **Gateways** operate on Layers 4 to 7

Repeaters amplify signals between two connectors. Both connectors must have the same physical properties. Repeaters can therefore only be used between similar connectors such as between two Ethernet cables or between two light-wave conductors. Among other uses, repeaters are also used in undersea cables. There are also **hubs**, which are repeaters with more than two ports. Hubs send incoming signals on each port (except of the input port) after amplifying them.

A **bridge** evaluates Layer 2 addresses and can thereby forward data units in a targeted way. In this way bridges also enable parallel communication and increase the performance of the network. A bridge with several or many ports is called a **switch**. Bridges and switches belong to the Data Link Layer, but as a required basic functionality they also implement the Physical Layer. Bridges and switches are presented in more detail in the chapter about the Data Link Layer.

A **router** uses Layer 3 addresses for routing. This means the router uses a routing table to determine to which port an incoming packet should be forwarded to. Routers belong to the Network Layer, but they also implement the Physical Layer and Data Link Layer. Routers are presented in more detail in the Chapter about the Network Layer.

A **gateway** operates at Layers 4 and 7 and is a special case that is not considered further in this course. It can, for example, be used for load balancing when an application is realized on different but similar servers. Then a query by one user can be answered by one server, and another user's query can be answered by another server.

important

Modern network components often cannot be assigned clearly to the layers. In particular there are devices that are primarily used as switches on the basis of Layer 2 addresses, but can also carry out routing tasks to a limited extent. In this course, however, the layers are treated separately from each other in order to clarify the differences.

2.8 Exercises - OSI Reference Model

task

Tasks for beginners

Task 1:

In the following chapters the tool Wireshark is used in several occasions so that you should get to know the tool already at this point. Download the tool from wireshark.org and watch an introduction video. Record some data traffic, e.g., during e-mail retrieval or Web surfing, and answer related questions afterwards.

Task 2:

Protocols which are used in reality should be mapped to the Hybrid Model. Find out on your own to which layers the protocols IPv6 and SMTP belong. They will be presented in details in following chapters. Please note that this task is about the Hybrid Model and not about the OSI Model.

Tasks for advanced learners

Task 1:
In this task you should get more familiar with Wireshark and answer additional questions about it.

Task 2:
The idea of this task is that you should at least once take a look at the real OSI Model standard document. Download it from the ISO web site and answer two questions about it.

2.9 Summary - OSI Reference Model

In this chapter, you have got a first impression of the complexity and variety of operation in computer networks, which become easier to handle using an appropriate layered model.

The principles of layer design and the resulting OSI Model should be familiar enough to you that you can explain the model, its seven layers and the respective tasks of these layers. You should also be aware that the actual situation in practice is better described with the Hybrid Model.

In addition, the basic possibilities of interaction between the layers via services (vertical communication) and within the layers via protocols (horizontal communication) have also been explained. In the subsequent chapters we will be able to build on and continue to deepen the knowledge gained here.