Prologue

W&T connects

We look back on over 40 years of experience in the development and production of microcomputer technology. During this time network and sensor technology as well as IT security have been added to our area of expertise. Our blue boxes ensure that your devices, switching and control signals as well as sensor data are safely and reliably in your network

www.wut.de

For 15 years, you have been able to find basic knowledge about network technology in this now well-established handbook, which is now in its eighth revised and expanded edition.

in this handbook we hope to give you the necessary technical orientation you may need. Here you will find complicated things simply.

We wish you all the best when reading and using this handbook, success with all your projects and a well connected future.

Best regards,
Rüdiger Theis,
Frank Thiel
and all WuT Employees

Rüdiger Theis

Frank Thiel

Do you have questions about our products or specific applications?

Our technicians will be happy to assist you: +49 202 2680-110

Content

Introduction

Computer or data networks allow an unlimited number of network nodes to exchange data with each other via a common infrastructure.

Communication data

When we speak of data networks, the first thing to do is to define what data are and how they are encoded.

No matter whether text, web pages, pictures, music, videos or other data are to be transmitted - a certain amount of bytes is always transported from A to B..

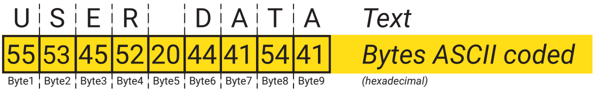
Bits and Bytes

At the lowest level, computers work with bits, i.e. with memory locations that can have the value 1 or 0. Eight bits form a byte.

A byte is a numerical value between 0 and 255. In data technology, bytes are usually represented in two-digit hexadecimal notation - i.e. 00 to FF (see chapter Number Systems).

Coding

Depending on the application, for example, a text becomes a certain amount of bytes, with each byte corresponding to one character. The assignment of which character corresponds to which numerical value is defined in the ASCII table (ASCII = American Standard Code for Information Interchange).



U	S	E	R		D	A	T	A	*Text*
55	53	45	52	20	44	41	54	41	*Bytes ASCII coded*
Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	*(hexadecimal)*

In a picture, a set of bytes would encode which color a particular pixel has in a particular position.

The significance of the individual bytes in the application is irrelevant for the transport it self. Here it is only an amount of bytes, i.e. numbers with which you can perform arithmetic operations if necessary.

Operating principle of a networks

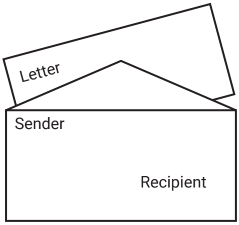
Basically all network topologies have one thing in common:

Each network member receives (at least) one own and unique address.

The user data to be transmitted are packed into a frame consisting of, for example, the address of the recipient, the address of the sender and the checksum in a „data packet“. In summary, the set of rules according to which such a frame is constructed is also called network protocol or protocol.

With the help of the address information, the user data in the resulting data packets can be transmitted to the correct recipient via shared lines.

It is no different with a letter: You put the letter in an envelope on which the recipient and sender are noted. The postman then knows to whom he should deliver the letter; the recipient can read where it comes from and to whom he can reply if necessary.



When transferring data within a network, the recipient also has the option of checking the completeness and accuracy of the received user data with the help of the checksum that is also sent.

On its way from one application to another, the data pass through various protocol layers. Each of these layers takes on a different function, on to which the next higher layer will build.

The lowest layer is the physical network access. In local networks, the various Ethernet standards are common here. We will see later how all information for the higher layers is actually transmitted in the data packets of the lowest layer.

If the Ethernet data packet is to be sent to a foreign network, it is addressed and transported by higher-level protocols such as TCP/IP.

TCP/IP delivers the data packet not only to the correct recipient, but also to the correct application. For this purpose, another higher-level protocol is usually used, which works together with the corresponding application program. For example, you receive an e-mail via the POP3 protocol and can retrieve it with your e-mail program.

Physical transmission

Different physical crosslinking technologies are available depending on the application. For local area networks, Ethernet is the most common network standard today; as early as 1996, approx. 86% of all existing networks were implemented using this technology. At that time, networks were used almost exclusively in office environments. In the meantime, Ethernet has also established itself in many places in the industrial environment and is increasingly replacing the previously common transmission methods such as serial field buses in factory buildings.

Local networks with Ethernet

Ethernet in its original form is standardized in the IEEE standard 802.3. Put simply, Ethernet uses various algorithms to transmit data in standardized packets to the network's members. Each participant has a unique address.

Original Ethernet standards

Over the course of time, various Ethernet variants have emerged, which can be distinguished significantly in terms of transmission speed and the cable types used. Originally, Ethernet was operated with a transmission speed of 10 Mbit/s; there were three different basic models, which are no longer of any significance today and which are at best still present in old installations:

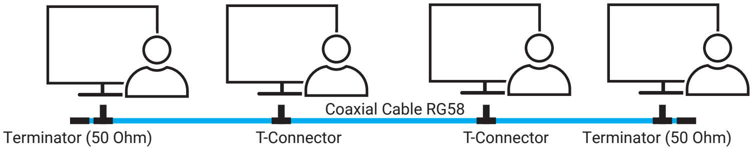
10Base5

Also often referred to as "Yellow Cable"; represents the original Ethernet standard and is no longer of any significance today. A finger-thick, inflexible and mostly yellow coaxial cable was used; the range was 500m.

10Base2

10Base2 is no longer used for new installations today and is only rarely found in older network installations.

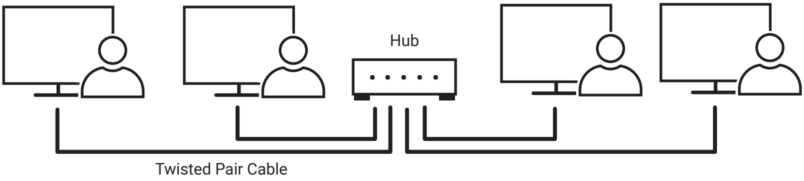
10Base2 is also known as Thin Ethernet, Cheapernet or simply as a BNC network. All network members are connected in parallel to a coaxial cable (RG58, 50 Ohm characteristic impedance). The cable must be terminated at both ends with a 50 Ohm terminator (terminal resistance).



If several devices share one common line, this is called bus topology. The disadvantage of this technology is its high susceptibility to interference. If the RG58 cabling is interrupted at any point, network access for all connected network participants is disrupted.

10BaseT

Each network node is connected via its own twisted pair cable to a so-called hub (star distributor), which forwards all data packets equally to all network nodes.



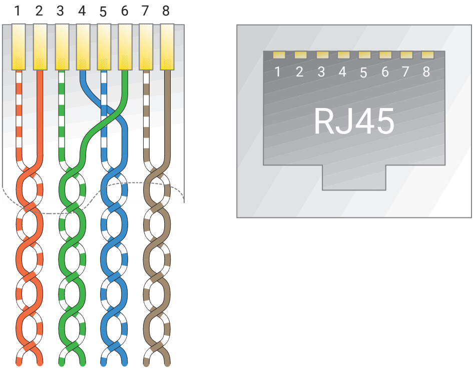
Even if 10BaseT physically operates in a star configuration, the bus principle is retained in terms of logic, as all connected network stations receive all network traffic.

Current Ethernet standards

Current wired Ethernet networks as well as 10BaseT use twisted-pair cables. The cables used originally come from American telephone technology. Twisted pair means that the cable wire pairs used for each signal are twisted together. Cables with four wire pairs are commonly used.

The RJ45 connectors used also originate from American telephone technology. The initially somewhat strange-looking division of the individual pairs and their coloring is laid down in AT&T Standard 258. 10BaseT uses only pins 1 and 2, as well as 3 and 6.

Twisted pair cables for network technology are categorized according to their transmission properties. Category 3 (Cat 3) cables were required for 10BaseT. For current networks, cables are used that correspond at least to category 5. The maximum permissible cable length between two active components is 100m.



100BaseT

With increasingly larger data volumes, Fast Ethernet with a transmission speed of 100Mbit/s was introduced in the 1990s.

As with 10BaseT, the cabling of the network participants is carried out via twisted pair cables. Switches are used as star distributors instead of hubs. Switches filter the data traffic so that each connected network member only receives the data intended for him (more about switches on the next page). As already mentioned, the cables used must be at least category 5 (Cat. 5). The maximum cable length is 100m.

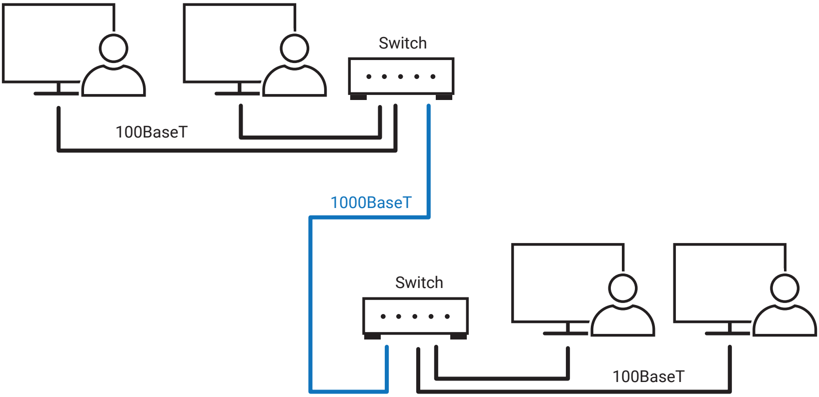
1000BaseT - Gigabit-Ethernet

The next Ethernet standard that will allow transmission speeds of one Gigabit (1000 Megabit) per second is 1000BaseT. To achieve this high bit rate, 1000BaseT uses a special data coding method.

The cabling requirements are the same as for 100BaseT. However, all four wire pairs of the twisted pair cables are used in parallel, which must be at least Category 5. 1000BaseT can be operated over max. 100m.

Initially, Gigabit Ethernet was mainly used as background cabling between switches. Such faster higher-level connections are also known as the backbone.

Today's PCs generally have a Gigabit Ethernet connection, so a direct, fast connection is nothing unusual anymore.



10GBaseT

In the meantime, transmission rates of up to 10 Gigabit/s are possible via twisted pair cables. However, 10GBaseT technology requires special network cards and infrastructure components and is therefore currently only used for the direct connection of servers or as a backbone. Distances of up to 100m are also possible with correspondingly high-quality cable (min. Cat. 6, better Cat. 7).

Switch and Hub

When 10BaseT became the physical standard for Ethernet networks, only hubs were initially used as star distributors. As described above, hubs forward the entire network traffic to all connected network nodes.

When 100BaseT later became the new standard, the hubs were equipped with autosensing ports. Autosensing means that the Ethernet port automatically detects the speed at which the connected terminal device is operating. Both interfaces involved then agree on whether 10BaseT or 100BaseT is used.

In the meantime, only switches are used instead of hubs. Switches no longer forward the entire Ethernet data traffic to all connected network participants. Instead, switches filter the data stream in such a way that only the data intended for the network node connected to that port is output.

The advantage of this technology is that the full bandwidth of the network connection is available to the individual connections. This increases the speed of data transmission for the network participants.

In addition, the autosensing capabilities of switches today usually include 1000BaseT.

Special physical Ethernet standards

In addition to the conventional cabling variants presented so far, there are now further possibilities for connecting nodes to a network.

PoE - Power over Ethernet

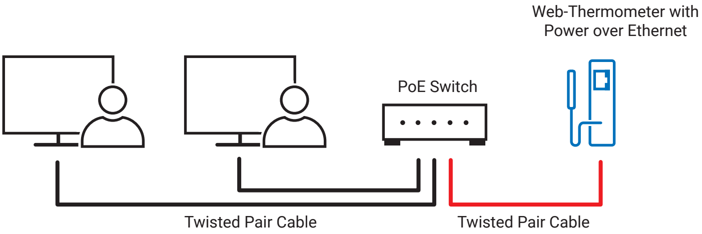
When talking about network participants, most people think of a PC first. Every stationary PC needs at least one more cable for power supply - usually 230V - besides the network cable. However, there are also network participants such as the W&T Web-Thermometers, which are considerably smaller than a PC and require relatively little power supply.

With PoE, such devices can be additionally supplied with power via normal Ethernet cabling. In order for this to work, the Ethernet interface of these devices has been technically extended accordingly. In addition, special switches or PoE injectors are required for operation, which feed the required energy into the network cable.

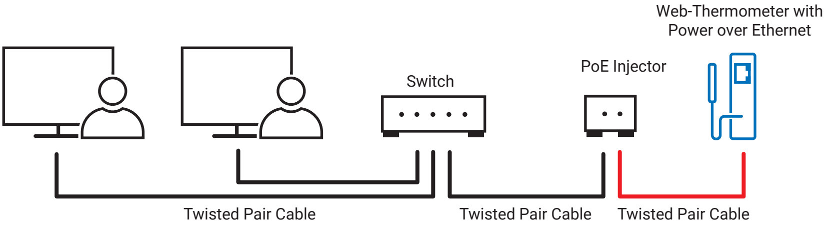
PoE supplies the end devices with 48V and currently knows five different power classes, which differ in the maximum power consumption. Using a special coding method, the PoE switch detects whether an attached device is PoE-capable or not, and switches the supply power on only if the required power can be provided an only when it is needed.

Class	Max. power supply	Withdrawal - rating	Example for end devices
0	15,4 W	0,44 W - 12,95 W	Individual devices
1	4,0 W	0,44 W - 3,84 W	W&T Web-IO a. Com-Server
2	7,0 W	3,84 W - 8,49 W	IP-Telephony
3	15,4 W	8,49 W - 12,95 W	Security camera
4	25,5 W	12,95–25,50 W	Panel PC

Thus, normal Ethernet components and PoE devices can be operated mixed on the same switch.



When the PoE supply comes from a switch, it is called an end-span solution. However, in existing networks, PoE devices can also be supplied with power via an intermediate PoE injector.



This case is called mid-span solution.

Ethernet via fiber optic cable

For cable lengths over 100 meters or in environments with strong electromagnetic interference, the transmission technology of 100/1000BaseT and 10GBaseT reaches its limits.

In data transmission via optical fiber (FO), the Ethernet data is converted into light signals and forwarded via a glass fiber. This has the advantage that there is no electrically conductive connection via the FO cable. Particularly in the case of cross-building connections, transmission via FO offers optimum protection against damage caused by thunderstorms.

A basic distinction is made between two physical fiber optic types:

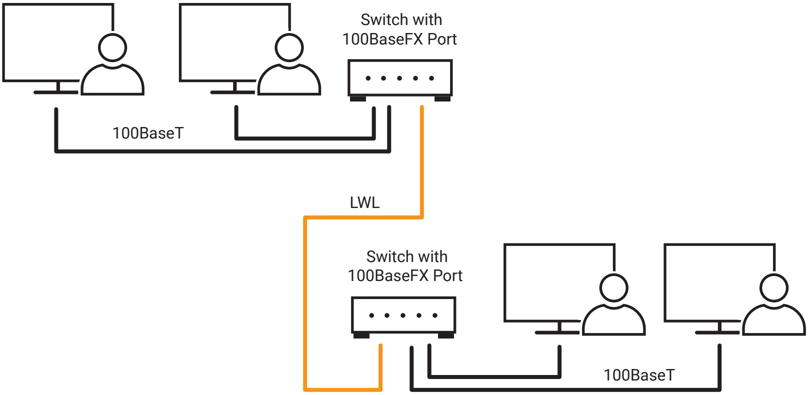
- **Multimode fiber**
  Using multimode fibers, distances of up to 2 km can be bridged.

- **Single-mode fibers**
  Another common term for single-mode is monomode. The processing and assembly of single-mode fibers is much more complex than for multimode fibers. However, depending on the transmission system, distances of up to 40 km can be bridged.

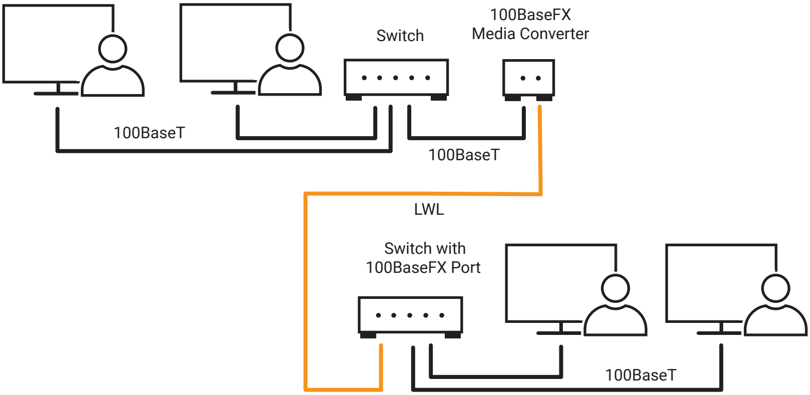
A detailed description of the different fiber optic standards can be found in the network ABC.

Fiber optic topologies

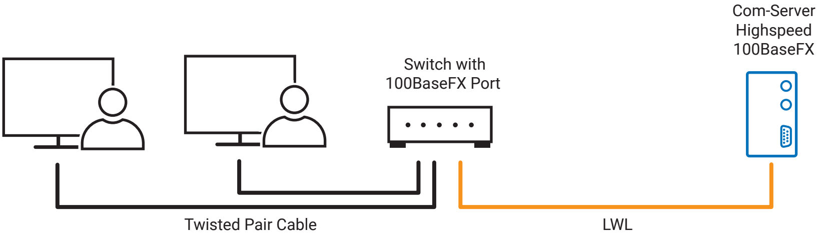
Since the costs for a fiber optic installation are significantly higher than those of 100BaseT, usually only certain parts of a network are designed as fiber optic. For example between switches as a backbone connection.



If one of the components used does not have a fiber optic connection by default, then a corresponding media converter can be used.



However, there are also end devices that are already equipped with a fiber optic port.

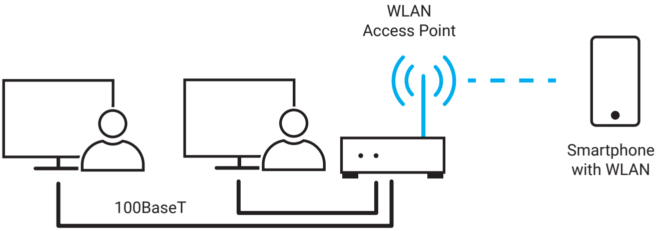


For example the W&T Com-Server Highspeed 100BaseFX. Such solutions are known as “Fiber to the Desk”.

Wireless LAN

WLAN realizes the network connection via radio waves and thus provides the user with independence from cables and thus mobility.

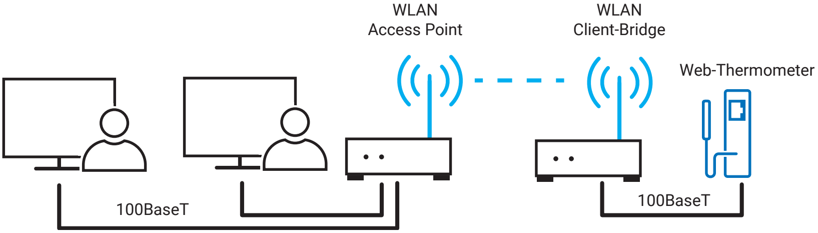
In general, a WLAN consists of at least one access point and one WLAN client.



The access point assumes the role of a star distributor. WLAN clients can log on to the access point and then communicate wirelessly with the rest of the network.

In most cases, access points are integrated into DSL routers or switches and act as a connection to a wired network.

Network participants without an integrated wireless LAN interface can access the WLAN via a WLAN client bridge. The client bridge acts as a media converter between the wireless and wired network.



The range of a WLAN can theoretically be up to 300 meters, depending on the environment and the components used. Within buildings, typical values of 25 meters are specified, although ceilings and walls can additionally limit the range.

Since the local extensions of radio networks can overlap, there are several possible channels (transmission frequencies). In the case of several WLANs at one location (not uncommon in apartment buildings or business premises), an unused channel should, if possible, be located between two channels in use to avoid mutual interference.

Another aspect of WLAN is data security. Radio signals can be received by anyone who is within range of the WLAN if the appropriate technical equipment is available.

In order to protect radio networks from unauthorized use and “eavesdropping”, the data is encrypted. A regular WLAN node must use both the encryption method used and the correct key to gain access to the wireless network.

Ethernet standards at a glance

Ethernet-Standard	Transmission medium	max. distance	Data rate	
10Base2	50 Ohm Coaxial cable	185 m	10 Mbit/s	*
10Base5	50 Ohm Coaxial cable	500 m	10 Mbit/s	*
10BaseT	100 Ohm TP cable cat.3	100 m	10 Mbit/s	
100BaseT	100 Ohm TP cable cat.5	100 m	100 Mbit/s	
1000BaseT/Gigabit	100 Ohm TP cable cat.5	100 m	1000 Mbit/s	
10GBaseT	100 Ohm TP cable Kat. 6 u. 7.	100 m	10 Gbits/s	
100BaseT-PoE	100 Ohm TP cable cat.5	100 m	100 Mbit/s	
1000BaseT-PoEt	100 Ohm TP cable cat.5	100 m	1000 Mbit/s	
100BaseFX	Multimode FO	2000 m	100 Mbit/s	
1000BaseSX	Multimode FO	550 m - 1000 m	1000 Mbit/s	
1000BaseLX	Multimode FO Single-mode FO	550 m 5 km	1000 Mbit/s	
WLAN 802.11a	Radio 5GHz	typically 25 m	max. 54Mbit/s	*
WLAN 802.11b	Radio 2,4GHz	typically 25 m	max. 11Mbit/s	*
WLAN 802.11g	Radio 2,4GHz	typically 25m	max. 54 Mbit/s	*
WLAN 802.11n	Radio 2,4GHz and 5GHz	typically 25m	max. 600 Mbit/s	*
WLAN 802.11ac	Radio 25GHz	typically 25m	max. ca. 6900 Mbit/s	*

** Here the network participants must share the maximum data rate. With the other standards, the specified data rate is available to every network node if it is connected to the network via a switch.*

The table lists only the most important fiber optic standards suitable for local networks. A complete overview can be found in the network ABC.

Combining different Ethernet standards

All Ethernet standards can be combined or mixed using appropriate infrastructure components.

For example, different parts of a building can be connected to each other via fiber optic cabling. Corresponding switches take over the conversion to 100BaseT or

Gigabit and can even provide the PoE supply if required. WLAN-enabled devices can be connected to the network via an access point or WLAN router.

The Ethernet data format

Whichever basic physical model is used, the logical structure of the data packets used is the same for all Ethernet topologies.

The Ethernet data format

The Ethernet address, also known as MAC-ID or Node-Number, is unchangeable. It is "burned" into the physical Ethernet adapter (network card, print server, Com-Server, router ...) by the manufacturer. The Ethernet address is a 6-byte value that is usually written in hexadecimal notation. An Ethernet address typically looks like this: 00-C0-3D-08-27-8B.

The first three hex values indicate the manufacturer code, the last three hex values are usually assigned by the manufacturer consecutively.

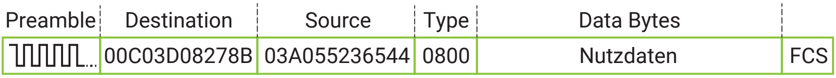
The Ethernet data packet

There are four different types of Ethernet data packets that are used depending on the application:

Data packet type	Application
Ethernet 802.2	Novell IPX/SPX
Ethernet 802.3	Novell IPX/SPX
Ethernet SNAP	APPLE TALK Phase II
Ethernet II	TCP/IP, APPLE TALK Phase I

Ethernet data packets of type Ethernet II are usually used in connection with TCP/IP.

Here is the structure of an Ethernet II data packet:



Preamble	Destination	Source	Type	Data Bytes	
...	00C03D08278B	03A055236544	0800	Nutzdaten	FCS

Preamble The bit sequence with continuous change between 0 and 1 is used to detect the beginning of the packet or synchronization.

	A collision (overlapping transmission of two stations) can be detected by a disturbed preamble. The end of the preamble is marked by the bit sequence “11”.
Destination	Ethernet address of the receiver
Source	Ethernet address of the sender
Type	Specifies the superordinate purpose (e.g. IP = Internet Protocol = 0800h)
Data Bytes	User data
FCS	Frame Checksum

The structure of the other Ethernet packets differs only in the fields “Type” and “Data”, which have a different function depending on the packet type.

The network participants process only those packets that are actually addressed to them.

Logical addressing and data transport

As a reminder: Each Ethernet address is burned into the corresponding terminal device by the manufacturer only once worldwide. This means that every terminal device in the network can be uniquely addressed.

In a group of several individual networks, however, the Ethernet address alone does not provide an indication of which network the station belongs to. Therefore, Ethernet alone is not sufficient for cross-network communication and the addressing required for this purpose.

In addition, Ethernet works connectionless: The sender does not receive any confirmation from the recipient whether a packet has arrived.

At the latest when an Ethernet network is to be connected to other networks, it is therefore necessary to work with higher-level protocols such as TCP/IP.

As early as the 1960s, the American military commissioned the creation of a protocol that would enable standardized information exchange between any number of different networks, regardless of the hardware and software used. This requirement led to the TCP/IP protocol in 1974.

Although TCP and IP are always mentioned in one word, they are two protocols based on each other. The Internet Protocol IP is responsible for the correct addressing and delivery of data packets, while the Transport Control Protocol TCP, which is based on it, is responsible for the transport and securing of data.

TCP/IP in the local network

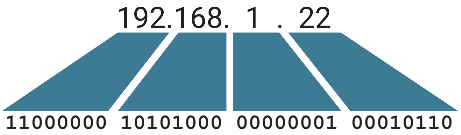
For the sake of clarity, we will first take a closer look at data transport and logical addressing with TCP/IP within a local network.

IP - Internet Protocol

To understand the addressing within a local network, we only need to look at the basic structure of the Internet Protocol IP and the Address Resolution Protocol ARP, which enables the assignment of IP addresses to Ethernet addresses.

IP-Addresses

Under IP, each network participant has a unique IP address, often referred to as an "IP number". This internet address is a 32-bit value, which is always given in the form of four decimal numbers (8-bit values) separated by dots for better readability (dot notation).



Each IP address must be unique in the entire connected network.

IP data packets

IP data packets also have a frame structure and contain a wealth of address and additional information in addition to the user data to be transported in the packet header. We will limit ourselves here to explaining the most important address information.

Structure of an IP data packet:

<table>
<tr><td>0 3</td><td>4 7</td><td>8 15</td><td colspan="3">16 31</td><td></td></tr>
<tr><td>VERS</td><td>HLEN</td><td>SERVICE TYPE</td><td colspan="3">TOTAL LENGTH</td><td rowspan="6">Header</td></tr>
<tr><td colspan="3">IDENTIFICATION</td><td>FLAGS</td><td colspan="2">FRAGMENT OFFSET</td></tr>
<tr><td colspan="2">TIME TO LIVE</td><td>PROTOCOL</td><td colspan="3">HEADER CHECKSUM</td></tr>
<tr><td colspan="6"><b>SOURCE IP ADDRESS</b></td></tr>
<tr><td colspan="6"><b>DESTINATION IP ADDRESS</b></td></tr>
<tr><td colspan="5">IP OPTIONS (IF ANY)</td><td>PADDING</td></tr>
<tr><td colspan="6"><b>DATA</b></td><td rowspan="3">Data Array</td></tr>
<tr><td colspan="6"><b>......</b></td></tr>
<tr><td colspan="6"><b>DATA</b></td></tr>
</table>

Source IP address:	IP address of the sender
Destination IP address:	IP address of the recipient

ARP – Address Resolution Protocol

In addition to the IP data packet, the IP driver also passes the physical Ethernet address to the Ethernet card driver. The IP driver uses the Address Resolution Protocol

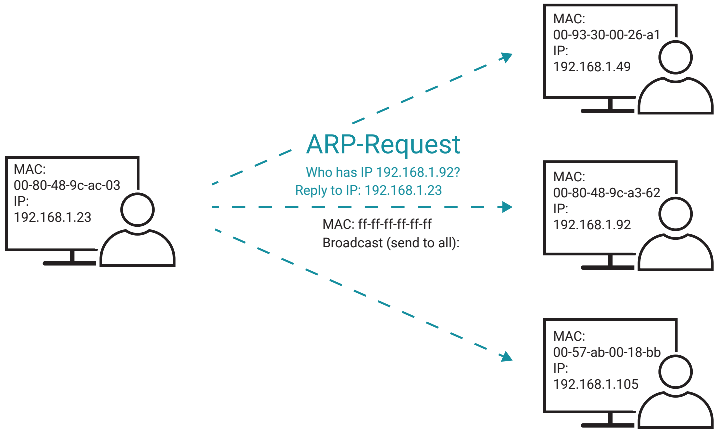
ARP to determine the Ethernet address of the receiver.

There is an ARP table in every TCP/IP capable computer. The ARP table is updated by the TCP/IP driver as required and contains the assignment of IP addresses to Ethernet addresses.

```
Internet Address    Physical Address     Type
192.168.1.23        00-80-48-9c-ac-03    dynamic
192.168.1.49        00-93-30-00-26-a1    dynamic
192.168.1.92        00-80-48-9c-a3-62    dynamic
192.168.1.98        00-c0-3d-00-1b-26    dynamic
192.168.1.105       00-57-ab-00-18-bb    dynamic
```

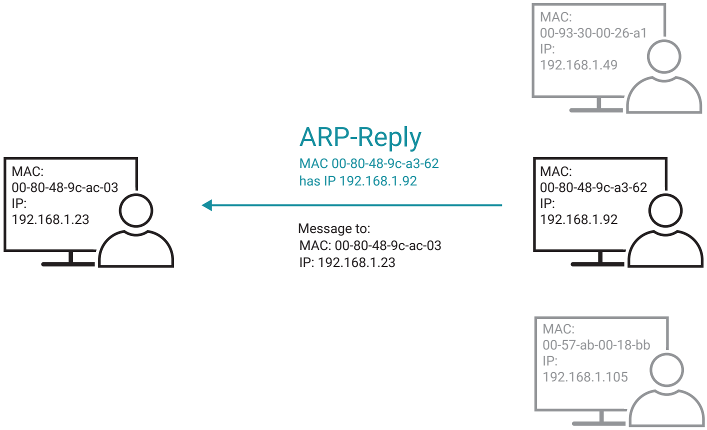
If an IP packet is to be sent, the IP driver first checks whether the desired IP address already exists in the ARP table. If this is the case, the IP driver passes the determined Ethernet address together with its IP packet to the Ethernet card driver.

If the desired IP address cannot be found, the IP driver starts an ARP request. An ARP request is a broadcast to all nodes in the local network.



To ensure that the broadcast call is acknowledged by all network nodes, the IP driver specifies FF-FF-FF-FF-FF-FF-FF as the Ethernet address. An Ethernet packet addressed to FF-FF-FF-FF-FF-FF-FF is always read by all network nodes. The desired IP address is specified as destination and the identifier for ARP is displayed in the Protocol field of the Ethernet header.

The network station which recognizes its own IP address in this ARP request confirms this with an ARP reply. The ARP reply is a data packet addressed to the ARP request sender on Ethernet level with the ARP identifier in the Protocol field. The data area of the ARP packet also contains the IP-Addresses of sender and receiver of the ARP reply.



The IP driver can now assign the Ethernet address taken from the ARP reply to the desired IP address and enters it in the ARP table.

Normally, the entries in the ARP table do not remain permanently. If a registered network device is not contacted for a certain time (in the case of Windows approx. 2 min.), the corresponding entry is deleted. This keeps the ARP table slim and allows the exchange of hardware components while keeping the IP address. These time-limited entries are also called dynamic entries.

In addition to dynamic entries, there are also static entries that the user creates in the ARP table. The static entries can be used to transfer the desired IP address to new network components that do not yet have an IP address.

W&T Com Servers also allow this type of IP address assignment: If a Com Server that does not yet have its own IP address receives an IP data packet addressed to it at Ethernet level, the IP address of this packet is evaluated and adopted as its own IP address.

Attention: Not all network components have this capability. For example, PCs cannot be configured in this way!

TCP - Transport Control Protocol

The question of how data should be transported is solved by transport protocols, each of which meets different requirements.

Because IP is an unsecured, connectionless protocol, it often works together with the TCP that is set up. TCP handles the secure delivery of the user data. TCP also establishes a connection between two network participants for the duration of the data transmission. When establishing the connection, conditions such as the size of the data packets are defined which apply for the entire duration of the connection.

TCP can be compared to a telephone connection. Subscriber A dials subscriber B; subscriber B accepts the connection by lifting the handset, which then remains connected until one of the two ends it.

TCP works according to the so-called client/server principle:

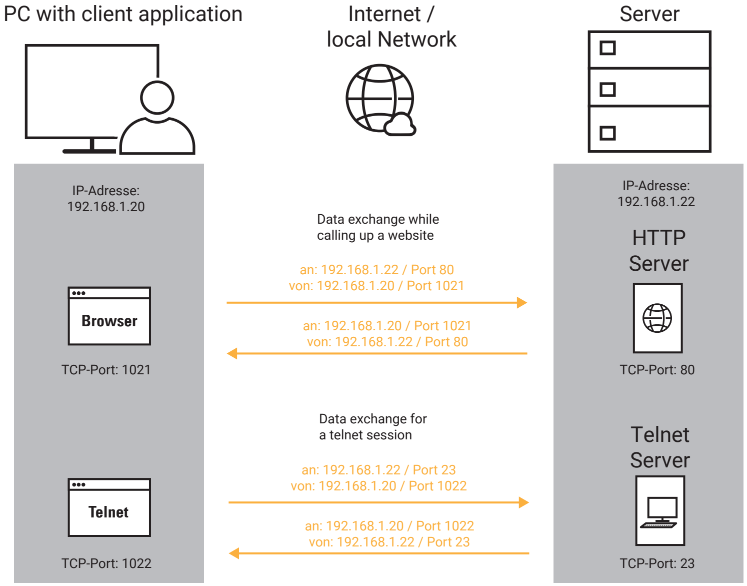
The network who establishes a connection (i.e. who takes the initiative) is called the client. The client uses a service offered by the server, whereby, depending on the service, a server can also serve several clients simultaneously.

The network participant to which the client establishes the connection is called the server. A server does nothing on its own, but waits for a client to establish a connection to it. In the context of TCP, we speak of TCP client and TCP server.

TCP saves the transmitted user data with a checksum and assigns a sequence number to each transmitted data packet. The recipient of a TCP packet uses the checksum to check that the data have been received correctly. If a TCP server has received a packet correctly, an acknowledgment number is calculated from the sequence number using a predefined algorithm.

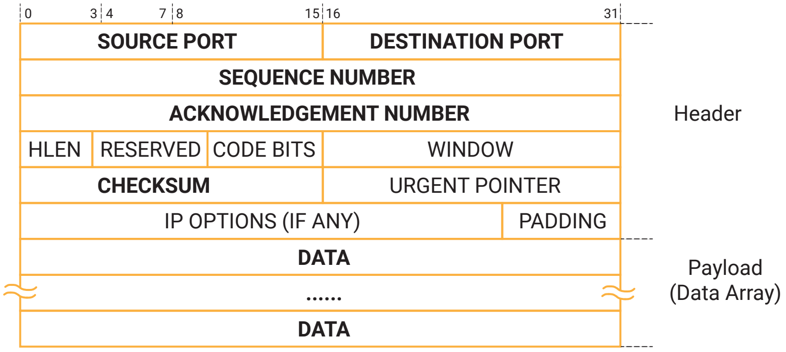
The acknowledgment number is returned to the client as an acknowledgment with the next self-transmitted packet. The server also provides its sent packets with its own sequence number, which in turn is acknowledged by the client with an acknowledgment number. This ensures that the loss of TCP packets is noticed and that they can be resent in the correct sequence if necessary.

Furthermore, TCP forwards the user data on the target computer to the correct application program. TCP uses port numbers, or “ports” for short. Different application programs - also called services - can be addressed via different port numbers. For example, Telnet can be accessed via port 23, HTTP, the service used to access web pages, can be accessed via port 80. If you compare a TCP packet with a letter to an authority, you can compare the port number with the room number of the addressed authority. For example, if the Road Traffic Office is located in room 312 and you address a letter to this room, you are also indicating that you want to use the services of the Road Traffic Office.


PC with client application
Internet / local Network
Server
IP-Adresse: 192.168.1.20
Browser
TCP-Port: 1021
Telnet
TCP-Port: 1022
Data exchange while calling up a website
an: 192.168.1.22 / Port 80
von: 192.168.1.20 / Port 1021
an: 192.168.1.20 / Port 1021
von: 192.168.1.22 / Port 80
Data exchange for a telnet session
an: 192.168.1.22 / Port 23
von: 192.168.1.20 / Port 1022
an: 192.168.1.20 / Port 1022
von: 192.168.1.22 / Port 23
IP-Adresse: 192.168.1.22
HTTP Server
TCP-Port: 80
Telnet Server
TCP-Port: 23

To ensure that the response from the target computer is returned to the correct location, the client application also has a port number. For PC applications, the port numbers of the client applications are assigned dynamically and independently of the type of application.

TCP also packages the user data in a frame of additional information. Such TCP packets are structured as follows:



Source Port: Port number of the sender's application

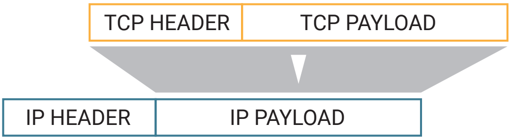
Destination Port: Port number of the recipient's application

Sequence No: Offset of the first data byte relative to the beginning of the TCP stream (guarantees that the sequence is maintained)

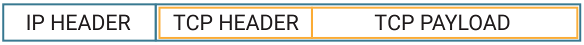
Acknowl. No: No. expected in the next TCP packet

Data: User data - payload

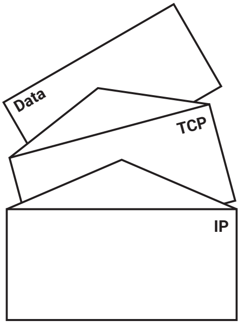
The resulting TCP packet is inserted into the Payload data area of an IP packet.



The IP packet then has the following structure:

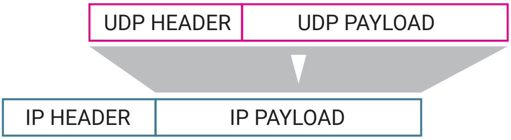


The payload is quasi put into an envelope (TCP packet), which is put into another envelope (IP packet).

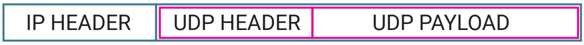


UDP – User Datagram Protocol

UDP is another transport protocol that is based on IP just like TCP.



The IP packet then has the following structure:



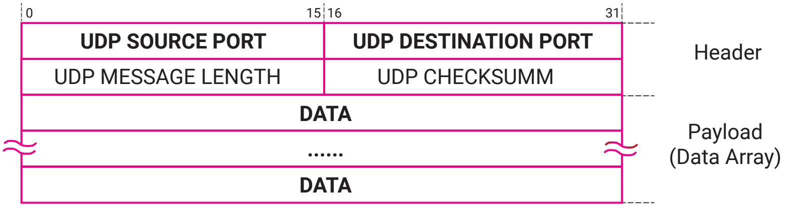
Unlike TCP, UDP works connectionless. This means that each data packet is treated as a single transmission and there is no feedback as to whether a packet has arrived at the recipient. UDP data packets are also called datagram.

With UDP there is no client/server principle. Both communication partners are called UDP peers.

However, UDP can be faster than TCP because no connections have to be established and closed under UDP and thus no timeout situations can occur: If a packet is lost, the data transmission is continued unhindered, unless a higher protocol provides for repetitions.

Data security under UDP must therefore be guaranteed by the application program in all cases.

UDP data packets are considerably smaller than e.g. TCP because they do not contain any data-securing information.



Source Port: Port number of the sending application (return port for receiver)

Destination Port: Target port to which the data is to be transferred at the recipient

As a rule of thumb one can say:

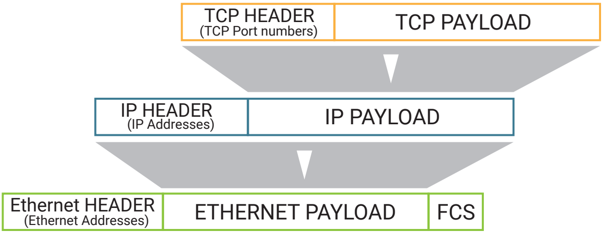
- For continuous data streams or large data volumes as well as in situations where a high degree of data security is required, TCP is generally used.
- The use of UDP is more effective with frequently changing transmission partners and a guarantee of data security through higher-level protocols.

The path of a character through the Ethernet

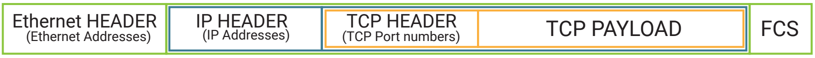
With TCP/IP (or UDP/IP) we have now learned the tools with which data is addressed and transported. In summary, the following section shows once again the path of a character through a local network.

TCP/IP is a purely logical protocol and always requires a physical basis. As already mentioned at the beginning, Ethernet is the most widely used physical network topology today. Thus, in most TCP/IP networks, Ethernet is also found as the physical basis.

TCP/IP and Ethernet are brought together by embedding each TCP/IP packet in the payload data area of an Ethernet packet.

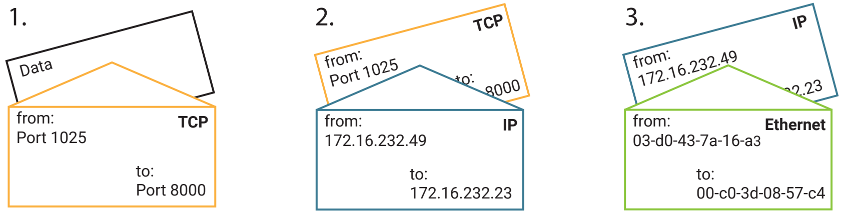


The complete package then looks like this:



The Payload (user data) passes several driver levels on its way from the application on the PC to the network:

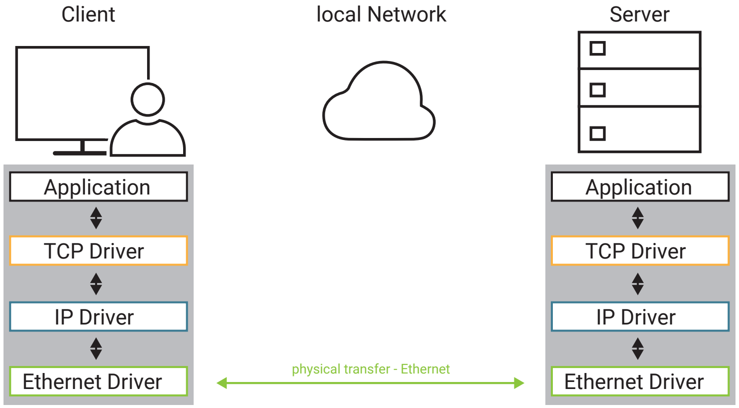
- **The application program passes the IP address configured by the user and the corresponding TCP port to the TCP/IP driver (often also called TCP/IP stack).**
- The TCP/IP driver coordinates the establishment of the TCP connection.
- The user data transferred by the application program are divided by the TCP driver into smaller transferable blocks depending on their size.
- Each data block is first packed into a TCP packet by the TCP driver (1.).
- The TCP driver passes the TCP packet and the IP address of the recipient to the IP driver.
- The IP driver packages the TCP packet into an IP packet (2.).
- The IP driver searches in the ARP table (Address Resolution Protocol) for the Ethernet address of the receiver specified by the IP address (if there is no entry, an ARP request is first triggered) and transfers the IP packet together with the determined Ethernet address to the Ethernet card driver.
- The Ethernet card driver packages the IP packet into an Ethernet packet and outputs this packet to the network via the network card (3.).



At the receiver the procedure takes place in reverse order:

- The Ethernet card recognizes by means of the destination Ethernet address that the packet is intended for the network node and passes it on to the Ethernet driver.
- The Ethernet driver isolates the IP packet and passes it on to the IP driver.
- The IP driver isolates the TCP packet and passes it on to the TCP driver.
- The TCP driver checks the content of the TCP packet for correctness and transfers the data to the correct application using the port number.

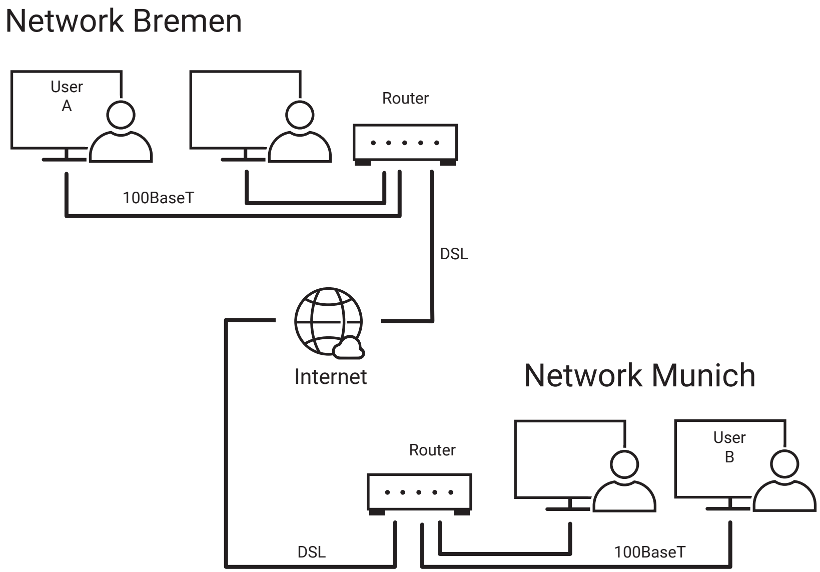
The example shows the interaction of logical addressing (TCP/IP) and actual physical addressing (Ethernet).



Only this interaction makes it possible to exchange data across networks and independent of hardware.

TCP/IP for cross-network connection

The Internet Protocol makes it possible to combine an unspecified number of individual networks into a complete network. It thus enables data exchange between any two network participants, each of whom is located in any individual network. The physical design of the networks or transmission paths (Ethernet, Token Ring, DSL, ...) is irrelevant.



The various individual networks are connected to each other via gateways/routers and thus merge into the internet or intranet. Addressing is still done via the IP address, which we will now take a closer look at.

Network classes

The IP address is divided into Net-ID and Host-ID, where the Net-ID is used to address the network and the Host-ID is used to address the network participant within a network. The Net-ID indicates whether the recipient, to whom the connection is to be established, is located in the same network as the transmitter. If this part of the IP address of the transmitter and the receiver matches, both are in the same network; if it does not match, the receiver can be found in another network.

Telephone numbers have a similar structure. Here too, a distinction is made between area code and subscriber number.

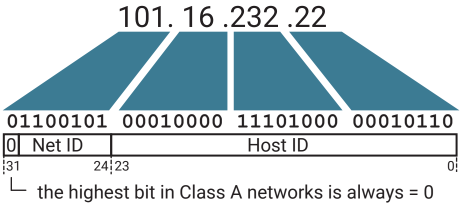
Depending on how large the share of the Net-ID in an IP address is, a few large networks with many nodes and many small networks with few nodes are conceivable. In the early days of the internet, the IP address space was divided into classes based on the size of the possible networks.

Class	Address	possible Networks	possible Hosts
Class A	1.xxx.xxx.xxx - 126.xxx.xxx.xxx	127 (2^{7})	appr. 16 Millions (2^{24})
Class B	128.0.xxx.xxx - 191.255.0.0	appr. 16000 (2^{14})	appr. 65000 (2^{16})
Class C	192.0.0.xxx - 223.255.255.xxx	appr. 2 Millions (2^{21})	254 (2^{8} - 2)

Two of the possible host addresses are omitted for the network address (e.g. 192.168.1.0) and the broadcast address (e.g. 192.168.1.255 - more on this later) of the network.

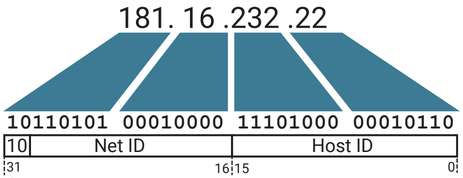
Class A

The first byte of the IP address is used to address the network, the last three bytes address the network node.



Class B

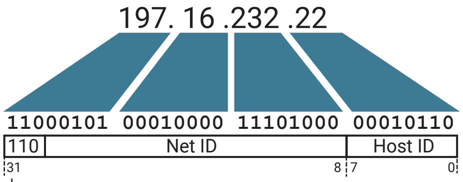
The first two bytes of the IP address are used to address the network, the last two bytes address the network node



the highest bits in Class B networks are always = 10

Class C

The first three bytes of the IP address are used to address the network, the last byte addresses the network node.



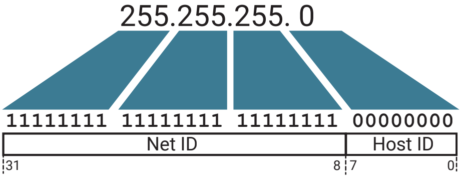
the highest bits in Class C networks are always = 110

In addition to the networks listed here, there are also Class D and Class E networks whose address ranges lie above the Class C networks. Class D networks and class E networks are of little significance in practice, since they are only used for research purposes and for special tasks. The normal internet user does not come into contact with these network classes.

Subnet-Mask

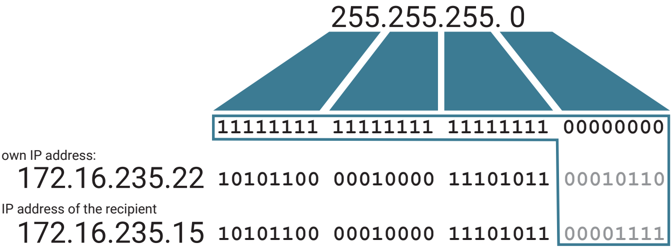
However, it is now possible to divide a network - regardless of the network class - into further subnetworks. To address such subnets, the Net-ID given by the individual network classes is not sufficient; one has to branch off a part of the Host-ID to address the subnets. In plain language this means that the Net-ID increases and the Host-ID decreases accordingly.

Which part of the IP address is evaluated as Net-ID and which as Host-ID is determined by the Subnet-Mask. The Subnet-Mask is exactly like the IP address a 32-bit value, which is represented in dot notation. If you look at the Subnet-Mask in binary notation, the part of the Net-ID is filled up with ones, the part of the Host-ID with zeros.



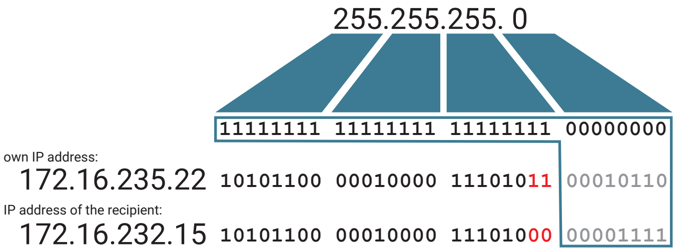
For each data packet to be sent, the IP driver compares its own IP-Address with that of the recipient. Here the bits of the Host-ID are faded out over the part of the Subnet-Mask filled up with zeros.

If the evaluated bits of both IP addresses are identical, the selected network node is located in the same subnet.



In the example shown above, the IP driver can determine the Ethernet address via ARP and pass it to the network card driver for direct addressing.

If even one of the evaluated bits differs, the selected network station is not in the same subnet.



In this case, the IP packet must be transferred to a gateway or router for further transfer to the target network. For this purpose the IP driver determines the Ethernet address of the router via ARP, even if the IP address of the desired network node is still included in the IP packet itself.

Gateways and Router

Gateways or routers are basically nothing more than computers with two network cards. Ethernet data packets received on card A are unpacked by the Ethernet driver and the contained IP packet is forwarded to the IP driver. The driver checks whether the destination IP address belongs to the subnet connected to board B and whether the packet can be delivered directly, or whether the IP packet is passed on to another gateway.

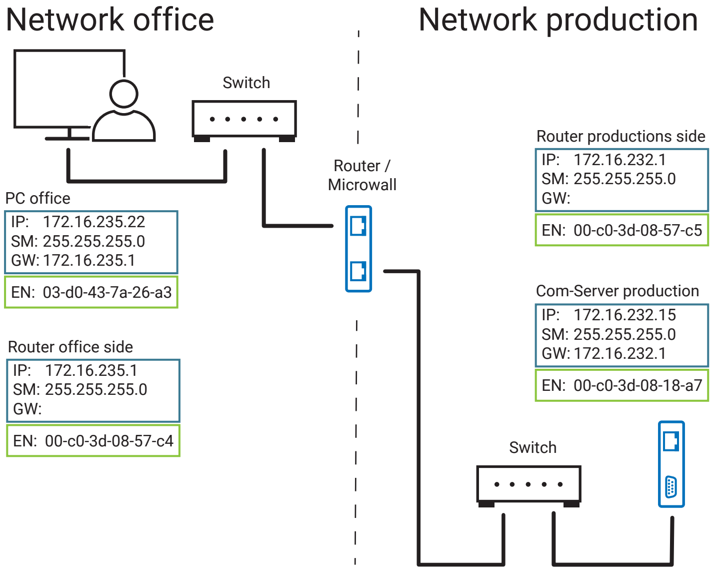
A data packet can pass several gateways/routers on its way from one network node to another. At IP level, the IP address of the receiver is entered on the complete route. On the Ethernet level, only the next gateway is addressed. Only on the section from the last gateway/router to the receiver is the Ethernet address of the receiver inserted into the Ethernet packet.

Besides routers that connect one Ethernet subnet to another Ethernet subnet, there are also routers that change the physical medium - e.g. from Ethernet to DSL. While IP addressing remains the same over the entire route, the physical addressing from one router to another is adapted to the physical conditions required on the part of the connection.

Between two DSL routers, the infrastructure of the corresponding internet provider operates. The physical addressing is then carried out, for example, via connection identifiers which ensure the unambiguous assignment of the respective DSL connection.

Routing - The path of data through several networks

The following section uses an existing Telnet connection to describe the path of a character over a routed network connection.



In our example we assume that a user in the office network has already established a Telnet connection to a W&T Com server in the production network; the connection between the networks is established via a suitably configured router.

The user at the PC enters the character “A” in the Telnet client application.

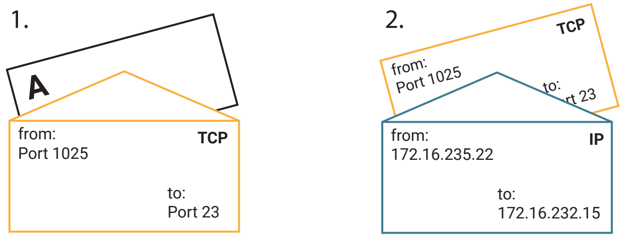
- The Telnet client program on the PC transfers the “A” to the TCP/IP stack as payload data part. The IP address of the receiver (172.16.232.15) and the port number 23 for Telnet were already transferred to the TCP/IP stack when the connection was established.

- The TCP driver writes the “A” into the payload data section of a TCP packet and enters 23 as the destination port (1.).

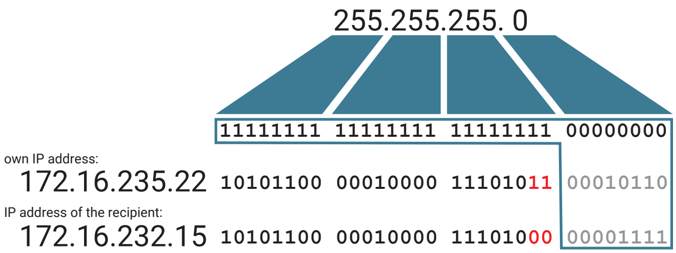
-

- The TCP driver passes the TCP packet and the IP address of the receiver to the IP driver.

- The IP driver packages the TCP packet into an IP packet (2.).



- The IP driver determines whether the IP packet can be delivered in its own subnet or to a router by comparing the Net-ID portions of its own IP address and the IP address of the recipient.



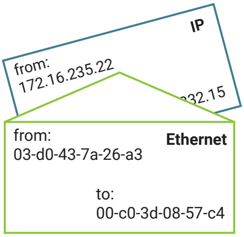
Here the Net-ID portions of the two addresses are not equal; the IP packet must therefore be transferred to the entered router.

- The IP driver determines the Ethernet address of the router via ARP. Since the TCP connection is already established, the IP address of the router will already be resolved in the ARP table.

```
Internet Address    Physical Address     Type
→ 172.16.235.1      00-c0-3d-08-57-c4    dynamic
  172.16.235.49     00-c0-3d-00-26-a1    dynamic
  172.16.235.92     00-80-48-9c-a3-62    dynamic
```

- The IP driver takes the Ethernet address of the router from the ARP table and passes it to the Ethernet card driver together with the IP packet.

- The Ethernet card driver packages the IP packet into an Ethernet packet and outputs this packet to the network via the network card.

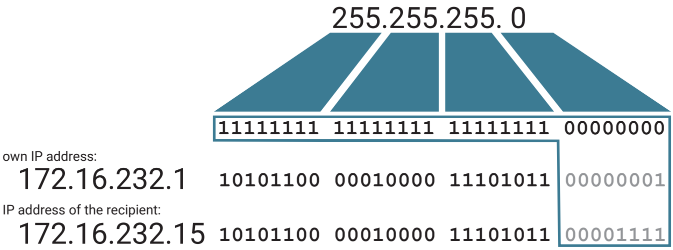


- The router takes the IP packet from the received Ethernet packet.

- The IP address of the receiver is compared with a so-called routing table. The router uses this routing table to decide whether it can forward the IP packet to the destination network. Depending on the network infrastructure, an IP packet may pass through several routers until it reaches the target network.

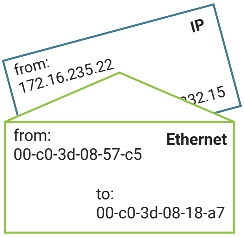
- For the Ethernet connection in the direction of the target network, the router uses IP

- Addresses and Subnet Mask determines whether the received IP packet can be delivered to the target network by one of the local Ethernet ports or must be passed to another router.



In our example, the IP packet has reached the target network and can be output at the corresponding Ethernet port and addressed via Ethernet.

- The router, which also maintains an ARP table internally to the target network, uses ARP to determine the Ethernet address matching the IP address and packages the IP packet, which is still unchanged in the addressing range, into an Ethernet packet.



- The Com-Server recognizes from the destination Ethernet address that the packet is destined for it and takes out the IP packet.

- The Com-Server's IP driver isolates the TCP packet and passes it on to the TCP driver.

- The TCP driver checks the content of the TCP packet for correctness and passes the data - in this case the "A" - to the serial driver of the Com-Server.

- The serial driver outputs the "A" on the serial interface.

With a TCP connection, the correct receipt of a data packet is acknowledged by returning an acknowledgment number. The acknowledgment packet passes through the entire transmission path and all associated procedures in the opposite direction. All this takes place within a few milliseconds.

VLAN - Virtual Local Area Network

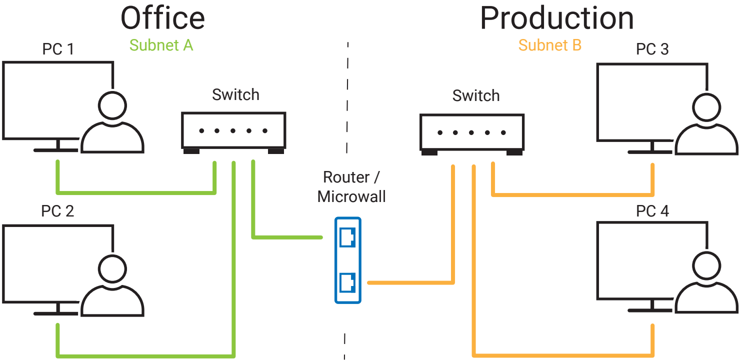
When larger networks are divided into smaller subnets, this is usually also associated with a spatial division of the subnets.

For example, in a company, the office area and production can each have their own subnets A and B, with the subnets being connected to each other via a router.

PCs and other end devices that are connected to a switch in the office area are thus exclusively connected to subnet A. Terminal devices connected in the production area belong to subnet B.

In addition to the logical separation, there is also a physical separation, which consists in the fact that all terminal devices connected to one switch belong to the same subnet.

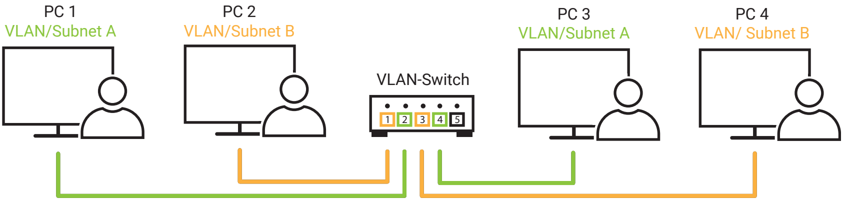
For example, if a PC in the office department is to be connected to the production subnet, a network cable would have to be laid from a switch belonging to the production subnet to the PC in the office.



By using VLANs, this rigid physical separation of the subnets is eliminated. Switches with VLAN support allow the network traffic of different subnets within the switch to be distributed to specific output ports. For this purpose, each port of the switch is assigned to a selected subnet. In this way, virtual subnets or VLANs are formed.

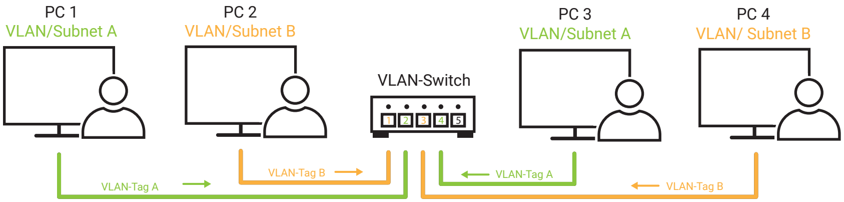
Port-based VLANs

With port-based VLAN, each port of a switch is assigned to a specific subnet or VLAN by appropriate configuration. Network participants connected to this port can only access the VLAN/subnet configured in this way.



Tagged VLANs

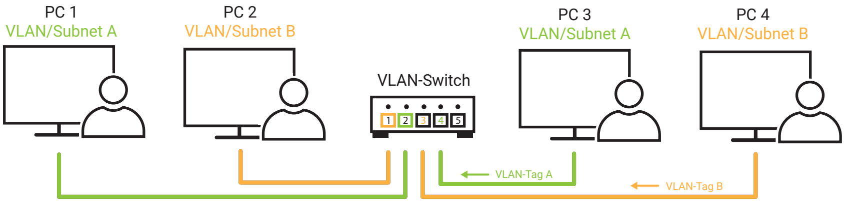
While with port-based VLAN the port/VLAN assignment is permanently stored in the switch, with tagged VLAN the connected network node determines to which VLAN he is connected. This is done via a tag that is sent in the header of the Ethernet data packet.



Tagged VLAN requires that the network hardware, the driver of the terminal device and the switch used all support tagged VLAN.

Port based and tagged VLAN in mixed operation

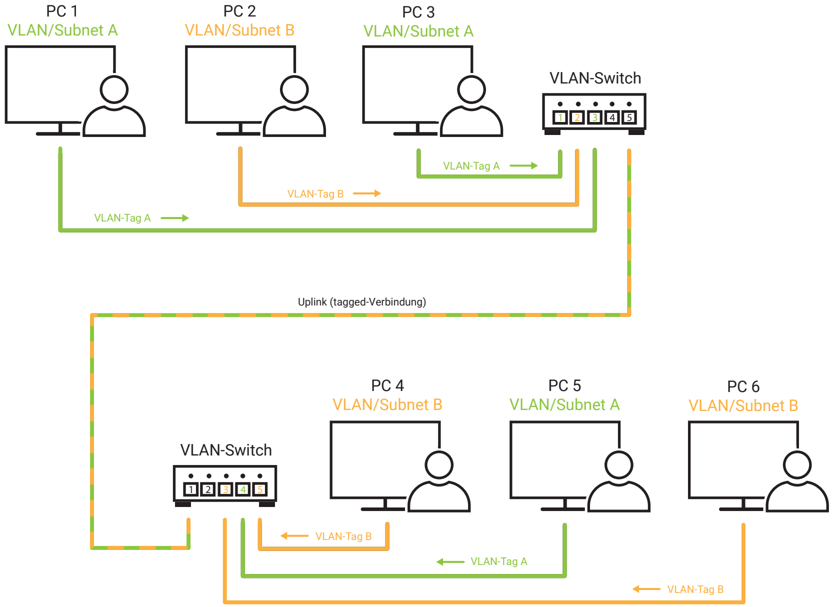
Most VLAN-enabled switches allow both variants in parallel, so that devices that do not support tagged VLANs can be connected port-based, while other devices use the tag to determine which VLAN they belong to.



VLANs over multiple switches

VLANs can also extend beyond more than one switch. With a purely port-based VLAN, however, a separate Ethernet cable is required between the switches for each VLAN, and the corresponding port on the switch must be configured accordingly for the VLAN.

In practice, the effort for double cabling is avoided and instead the tagged method is used to interconnect switches in the VLAN environment. The ports used for this purpose on the switches must be configured accordingly.

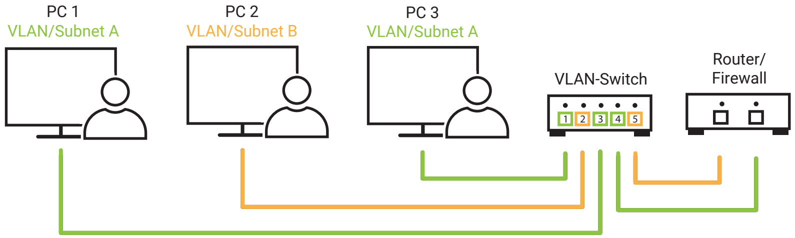


For the tagged connection between the switches it does not matter whether the individual network nodes are connected in a tagged or port-based manner.

More than one tagged connection can be used between two switches to achieve a better data throughput. This procedure is called trunking.

VLANs and Routing

If the users of two VLANs are to exchange data with each other, this is only possible by means of routing. For this purpose, a standard router can be connected to appropriately configured ports of the switch.



Alternatively, there are special switches that handle the routing between the VLANs internally themselves.

Protection by firewalls

The basic function of a router is limited to switching IP data packets from one network to another. This is done as shown by matching the receiving IP address, Net-ID and Subnet-Mask. If the requirements for routing are met, the data packets are forwarded unfiltered.

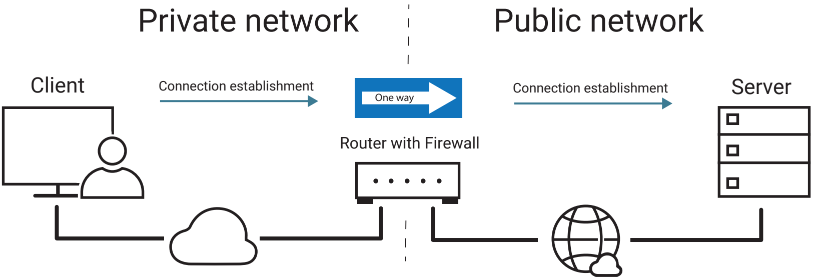
In networks connected to the internet via a router, anyone would have access to the local network from anywhere. This would of course be disastrous for data security.

That is why most routers also act as a firewall. Rules can be configured for the firewall, which determine which data packets are forwarded in which direction.

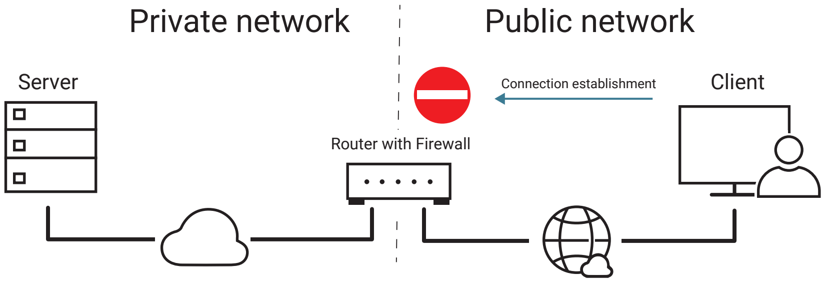
The configuration possibilities of firewalls are so diverse that they would provide enough material for a book in their own right. Therefore we will limit ourselves here to the most essential information.

The most commonly used firewalls are now routers that connect a local network directly to the internet via the provider's connection technology (e.g. DSL routers).

By default, such routers are configured so that connections from the local network to the internet are possible without restriction (client in the local network, server in the internet).



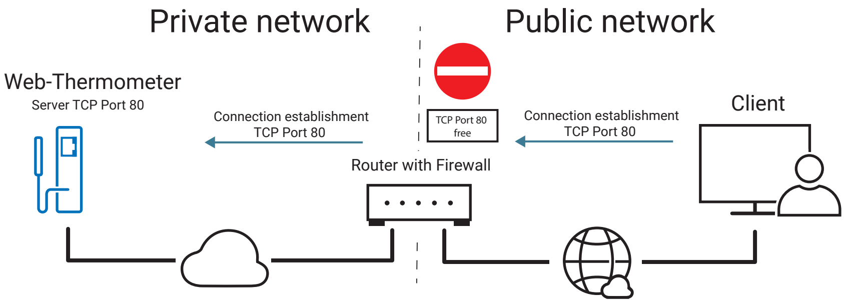
Connections from the internet to the local network are rejected by the firewall. (client on the internet, server on the local network).



However, there are also cases where certain server services in the local network should be accessible for clients from the public network.

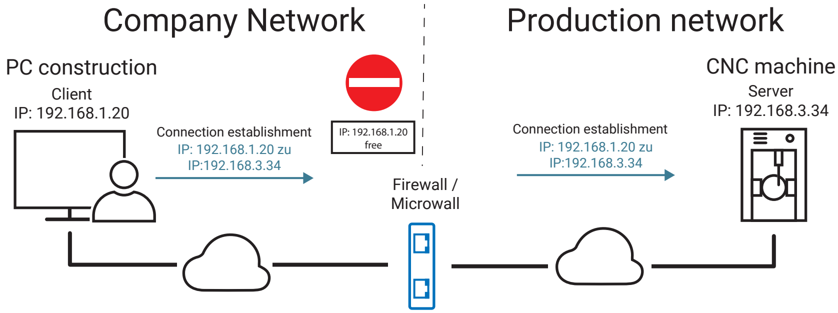
If, for example, the website of a W&T Web Thermometer is to be released for access from the internet, a corresponding rule can be configured:

"On port 80 (HTTP for browser communication), a TCP connection may be established from outside.".



In large company networks in particular, firewalls are also used to seal off subnetworks, for example to protect network-controlled CNC machines in production from external access. The configuration options for such firewalls usually go much further than simply blocking and opening certain ports.

For example, access from the general network into the subnetwork can be restricted to a specific network participant.



The rules can be fine-tuned even further. For example, approval can be given for combinations of IP address and port. IP address ranges can also be defined. Of course, a multitude of rules is possible for both connection directions.

Please note that the rules apply to the TCP connection setup.

Once the TCP connection is established, data can be exchanged in both directions.

Of course, rules for UDP datagrams can also be set up. Depending on the firewall, there are different possibilities for this.

Just a reminder: With UDP there is no connection and therefore no automatically defined return channel. Therefore rules for UDP communication often have to be configured for both directions so that the participant on the other side of the firewall can respond to a UDP datagram.

However, some firewalls also recognize the sender IP address and sender port and automatically release the return channel even with UDP.

NAT - Network Address Translation

If you want to connect a network with ten end devices to the internet via a normal router, e.g. using DSL, each of these end devices would need its own unique IP address.

As already mentioned, public IP addresses, i.e. those that are assigned once by IANA and can therefore be connected to the internet, are now in short supply.

In addition to these public IP addresses, however, there is an address space for private networks. Here, the term “private” stands for “non-public” and also includes company networks. Depending on the network size, these address ranges are provided:

for Class A: Networks 10.0.0.1 bis 10.255.255.254
for Class B: Networks 172.16.0.1 bis 172.31.255.254
for Class C: Networks 192.168.0.1 bis 192.255.255.254

Administrators can freely use these address ranges when setting up their private network. Since one and the same address can occur in several networks, addresses from these ranges are only unique within the own network. Therefore, no normal routing to these addresses is possible. This is exactly where NAT routing provides a remedy.

With NAT (Network Address Translation), a type of routing was created that allows a large number of participants in a private network to the internet to be represented by only one public IP address.

As a reminder: With normal TCP/IP data traffic, the IP address addresses the network participant, the port number addresses the application in the device.

With NAT routing, the port number is also used as additional address information for the terminal device itself.

Client in private network
The mode of operation of NAT routing will be explained here using a little example.

In a private Class C network the address space 192.168.1.x is used. A NAT router is used as a gateway to the internet, which operates externally with the IP address 197.32.11.58

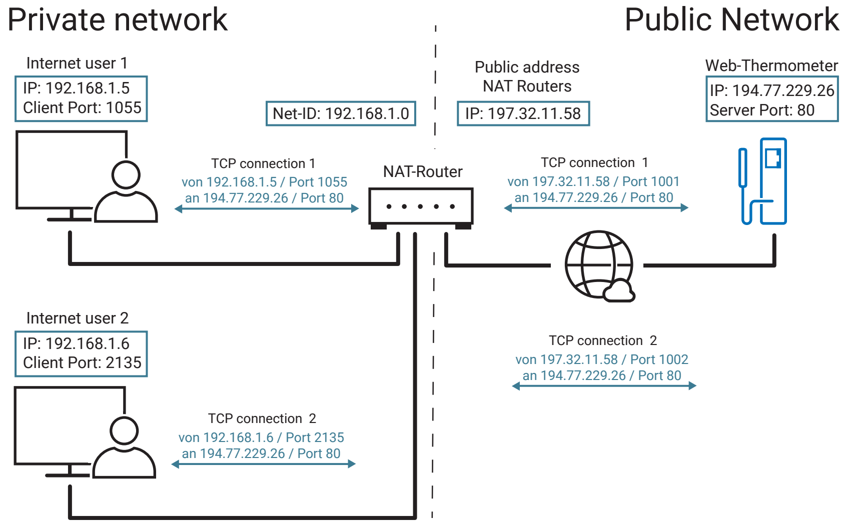
The PC with the network-internal IP address 192.168.1.5 establishes a TCP connection to the W&T web thermometer (IP 194.77.229.26, Port 80) on the internet using the local port 1055.

A second PC with the IP address 192.168.1.6 also establishes a TCP connection to the web thermometer and uses the local port 2135 for this purpose.

To be connected to the web thermometer, the PCs first contact the NAT router.

In the TCP/IP data packets that are forwarded to the web thermometer, the NAT router exchanges the IP address of the respective PC for its own public IP address.

The port number specified by the PC can also be exchanged for a port number managed by the NAT router.



The NAT router manages the assigned port numbers in a table, which is structured as follows:

from outside	in private network	
Port No.	corresponding IP	corresponding Port No.
1001	192.168.1.5	1055
1002	192.168.1.6	2135

The web thermometer therefore receives data packets for both connections in which the NAT router is entered as the sender. However, a separate port number is used for each connection.

The web thermometer inserts this “alternative” address information in all data packets in the direction of the two PCs. This means that the TCP/IP packets are structured in such a way that the NAT router is the recipient.

If the NAT router receives such a data packet addressed to it, it uses the assignment table to determine who the actual recipient is and replaces the received address data with the original network internal connection parameters.

The assignment table for outgoing connections (client in the private network, server outside) is managed dynamically and can of course contain much more than two connections. Thus, any number of connections can be routed outwards.

Server in private network

Data traffic in the other direction (server in the private network, client outside) can of course also be handled via NAT.

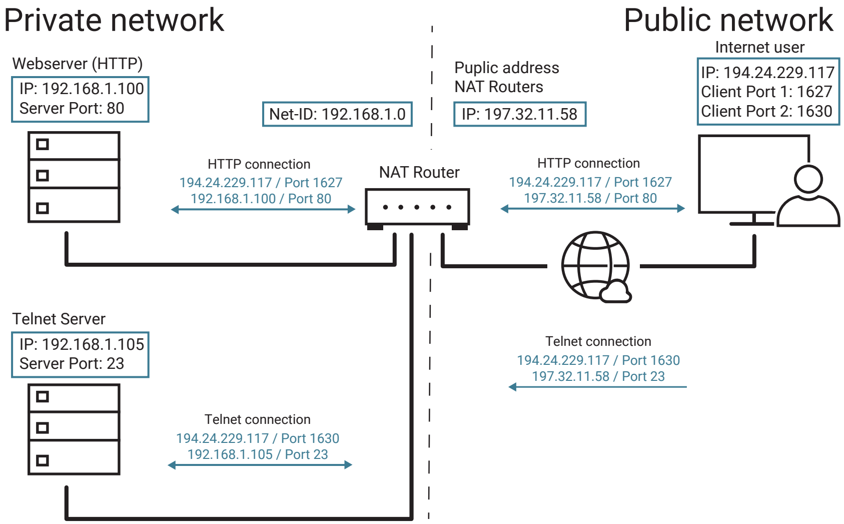
Here too, an assignment table is used to determine to which end device and port incoming connection requests or data packets are to be routed.

In contrast to the assignment table for outgoing connections, the server assignment table is static. The administrator must create a corresponding entry in the list so that servers in the private network can be reached from outside.

Every server service that needs to be accessible from public network requires a separate entry in the server list.

If, for example, a web server (HTTP = port 80) and a Telnet server (Telnet = port 23) are to be accessible from outside, the server table could look like this:

from outside	in private network	
Port No.	corresponding IP	corresponding Port No.
80	192.168.1.100	80
23	192.168.1.105	23



Detailed information on the Telnet and HTTP protocols follows in the chapters Application Protocols and Web Protocols.

The NAT router exchanges the connection parameters in the same way as for the connections shown in the previous section.

In a private network that is mapped to the internet via a NAT router with only one IP address, each server port may of course only appear once in the server table. This means that a special server service with a specific port number can only be offered by an internal terminal device.

Port Forwarding

Port forwarding can be considered an advanced form of NAT routing. While NAT routing retains the outward representation of the port on the private network, port forwarding also changes the port number.

Example: Two servers with the IP addresses 192.168.1.100 and 192.168.1.105 are operating in the private network. Both servers can be reached within the private network via port 80 as HTTP servers.

In addition, both servers should also be accessible from the internet. Of course, this cannot be done using the same port. The router must therefore represent at least one of the servers externally via a different port.

from outside	in private network	
Port No.	corresponding IP	corresponding Port No.
80	192.168.1.100	80
81	192.168.1.105	80

However, port forwarding is not supported by all routers.

Transmission protocols

In the previous chapters we have dealt with Ethernet as a transmission protocol, IP as a protocol for logical addressing and TCP/UDP as set up protocols for data transport or transport protection.

In addition, cross-network data exchange and the associated routing were described.

In times of the internet, however, routing is almost always linked to bridging the section between two networks using a technology other than Ethernet. For example with DSL, via the mobile phone network or other physical standards.

We will leave the physics used for this purpose out of consideration for the time being (You will find more about this in the chapter “The way to the internet”).

Regardless of the physical standard, however, a higher-level protocol is always required for remote data transmission (RDT), which performs the following tasks:

- Establishing a logical connection between the two locations
- Authentication (checking the access authorization)
- Preparation of incoming data traffic for transmission and restoration of the original data format at the end of the transmission path
- Data backup
- Encryption of the transmission data if necessary
- Termination of the logical connection after completion of the data transfer

SLIP - Serial Line IP Protocol

A first approach for the transmission of dial-up data was SLIP. SLIP is a very simple protocol that is only suitable for the transport of IP data traffic and does not meet all the requirements listed above.

With SLIP, the complete IP data packets are simply extended by a fixed start and end character. The sender replaces random characters of this type in the IP packet with a combination of replacement characters.

When they have been prepared in this way, they are then placed on the line packet by packet.

From the start/end characters of a packet, the receiver can tell where the actual IP packet begins and ends. The receiver then exchanges the replacement characters for the originals and removes the start/end characters.

Due to the limitation to IP data transmission and the lack of security mechanisms, SLIP is no longer used for normal internet access. However, where remote network segments are to be connected over distances that are no longer possible with normal Ethernet cabling, SLIP can still be a practical and inexpensive solution.

The W&T Com Servers can be configured, for example, as SLIP routers and thus transmit TCP/IP data via RS232 or RS422 cabling.

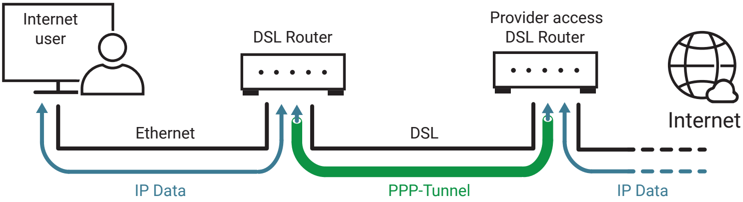
PPP - Point-to-Point Protocol

An essential feature of PPP is that in addition to IP data, data can also be transmitted in the form of other protocols such as IPX/SPX etc.

This means that data of any format must be exchanged unchanged between two network locations. This technique is also known as data tunneling.

Both for access to the internet and for connection to a remote, non-public network, PPP provides the necessary secure data transmission.

PPP creates a tunnel through the off-network environment, so to speak.



Protocol procedure

The establishment of a PPP connection takes place in several steps and requires an existing physical connection such as DSL:

1. Negotiating the connection options
   To determine which options PPP should work with,
   the LCP protocol (Link Control Protocol) is used.
   Negotiable are among others
   - Type of authentication
   - Block size of the transmission data
   - Data compression
   - Type of data to be transmitted (IP, IPX, ...)
2. Authentication
   Here User-ID and a password are transferred.
   There are two types of password transfer:
   - PAP - Password Authentication Protocol
     Password transfer readable in plain text
   - CHAP - Challenge Handshake
     encrypted password transfer
3. Configuration of higher-level network protocols
   If a connection to networks with higher protocols (e.g. Internet Protocol) is to be established via PPP, it is necessary to make certain settings concerning the corresponding protocol.
   The necessary information is transferred using the NCP (Network Control Protocol). In the case of internet access via PPP, the internet-specific IPCP (Internet Protocol Control Protocol) is used as the NCP. IPCP allows, for example, the assignment of an IP address for the duration of the PPP connection.
4. Transmission of the user data
   As soon as all connection options have been defined and the user has proven his access authorization, the actual exchange of user data begins.
   In the case of an internet connection, this can be data in the form of all IP-based protocols (UDP, TCP, Telnet, FTP, HTTP...).
5. Termination of the PPP connection
   The connection is also cleared via LCP

Protocol structure

Similar to Ethernet, PPP embeds the data to be transported in a defined packet structure:

1 Byte	1 Byte	1 Byte	1 oder 2 Bytes	n Bytes	2 Bytes	1 Byte
Flag	Address	Control	Protocol	Information	FCS	Flag

Flag (1 Byte)

Start character for packet synchronization or packet detection.

Address (1 Byte)

A point-to-point connection requires no addressing.

Nevertheless, this field is available for compatibility with other network protocols, but is not used by PPP and is arbitrarily filled with the value 255.

Control (1 Byte)

This field was originally intended for packet numbering, but it always has the value 3 in PPP, since the system works without packet numbering.

Protocol (1 or 2 Bytes)

The content of this field indicates how the current PPP packet is used: Connection setup, control information, authentication, data transport, connection termination, ...

Information (n Bytes)

At this point the actual information (e.g. IP data) is transmitted. For control packets, the control options are available here in LCP format.

The size of this field is negotiable via LCP, but is usually 1500 bytes. If the information to be transported is smaller, it is filled with fill characters.

FCS (2 Bytes)

Checksum to check the received data

Flag (1 Byte)

End marker for packet synchronization

The possibility of transmitting various independent IP services and protocols simultaneously via a PPP connection means that entire networks can also be connected via PPP. However, this requires suitable routers.

Auxiliary protocols

After the basic protocols of TCP/IP data transmission were explained in the previous chapter, the following section will deal with the application protocols that are based on these basic protocols.

The application protocols are divided into auxiliary protocols and actual application protocols. Auxiliary protocols are used for management and diagnostic purposes and often run in the background, invisible to the user.

Auxiliary protocols include:

- DHCP
- DNS
- DDNS
- DynDNS
- ICMP (Ping)

DHCP - Dynamic Host Configuration Protocol

As a reminder: Each Ethernet terminal device has a globally unique Ethernet Address (MAC address), which is unchangeably specified by the manufacturer. For use in TCP/IP networks, the network administrator also assigns the terminal device an IP address that matches the network.

If no DHCP is used, the IP addresses are assigned “classically”:

- For devices that allow direct user input (e.g. PCs), the IP number can be entered directly in a corresponding configuration menu.

- In the case of “black box devices” (e.g. Com Servers), there is on the one hand the ARP procedure via the network, and on the other hand there is the possibility of entering the configuration information via a serial interface. In addition, some manufacturers provide tools (e.g. the WuTility tool from W&T) to configure embedded devices directly from the PC.

In addition to the IP address, further parameters to be configured are the subnet mask and gateway as well as a DNS server (more on this in the next chapter). In large networks with many different end devices, however, this quickly results in a high degree of configuration and administration effort.

With DHCP the network administrator is offered a tool with which the network settings of the individual end devices can be configured automatically, uniformly and centrally.

To use DHCP, at least one DHCP server is required in the network, which manages the configuration data for a specified IP address range.

DHCP-capable end devices request their IP address and the associated parameters such as subnet mask and gateway when booting from this server. DHCP servers provide three basic options for IP address assignment and configuration:

- Allocation of addresses from an IP address pool
- Allocation of a reserved IP address
- Exclusion of certain IP addresses

Assigning the IP address from an address pool

A range of IP addresses is defined on the DHCP server from which a requesting network node is assigned a currently unused address. With this procedure, the allocation is usually limited in time, whereby the period of use (lease time) can be defined or completely deactivated by the network administrator. In addition, important data (lease time, subnet mask, gateway, DNS server, etc.) can be stored in a configuration profile that applies to all end devices served from the address pool.

Advantages

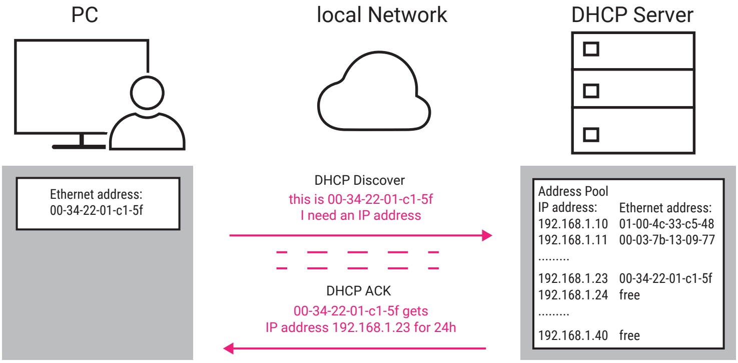
- Low administration effort
- Users can access the network at different locations with the same end device without configuration effort.
- If not all end devices are active in the network at the same time, the number of possible end devices can be greater than the number of available IP addresses.

Disadvantages

- A network node cannot be identified by its IP address because it is not possible to predict which IP address a terminal device will be assigned at startup.

Example: Typical cases for the allocation of IP addresses from an address pool are university networks. Here there are networks with an almost unlimited number of potential users, but only a few of them actually work in the network. Thanks to DHCP, students can take their notebook or tablet from one laboratory to another and operate it in the network without changing the configuration.

In order to keep the administration and configuration effort low, most home networks (a DSL router, a few PCs, printers and smart phones) also work with DHCP. The DSL router takes over the task of the DHCP server.



Address Pool

IP address:	Ethernet address:
192.168.1.10	01-00-4c-33-c5-48
192.168.1.11	00-03-7b-13-09-77
....	
192.168.1.23	00-34-22-01-c1-5f
192.168.1.24	free
....	
192.168.1.40	free

Allocation of a reserved IP address

The network administrator has the option of reserving individual IP addresses for specific end devices. For this purpose, the Ethernet address of the terminal device is assigned to the IP address on the DHCP server; an individual configuration profile can also be created for each reserved IP address.

Advantages:

- Despite individual configuration, all network settings can be transferred to a central location and do not have to be carried out on the end device itself.
- Terminal devices can be specifically addressed via their IP address.

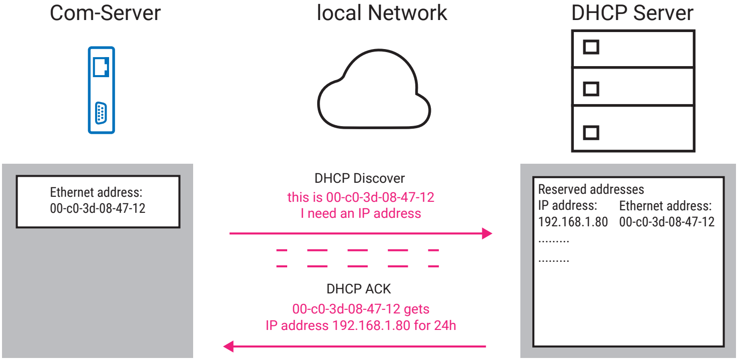
Disadvantages:

- Despite individual configuration, all network settings can be made at a central location and do not have to be made on the terminal device itself.
- Terminal devices can be specifically addressed via their IP address.

Example: Configuration of DHCP-capable end devices such as print servers or Com Servers, which require addressing via IP address depending on the application. In the DHCP manager, the Ethernet address of the corresponding terminal device is entered in the reserved IP address. In case of the Com-Server additional parame-

ters such as subnet mask, gateway (router) and DNS server can be entered.

It must be added that some end devices also use the older BootP protocol to request their configuration. BootP is a precursor of DHCP and is also supported by DHCP servers.



With older “black box devices” the BootP protocol can be used to force the transfer of a reserved IP address in all cases. If the DHCP server does not have an entry matching the Ethernet address of the com server, the BootP request should be ignored and the device will keep the currently setting for the IP address.

Unfortunately, not all DHCP servers handle it this way and some will also assign an IP address from the address pool in response to a BootP request.

Exclusion of certain IP addresses

For end devices that are neither DHCP- nor BootP-capable, the network administrator has the option of excluding individual IP addresses or even entire address ranges from being assigned by DHCP.

In this case, the configuration must either be carried out on the terminal device itself or by using the tools supplied.

Disadvantages:

- non-uniform and possibly decentralized configuration
- higher administration effort required

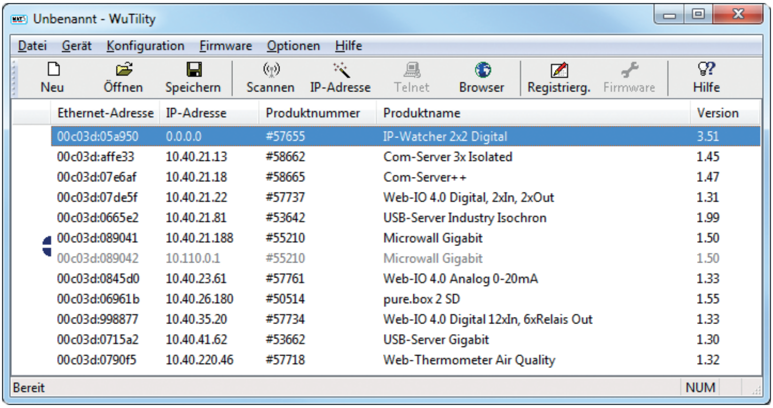
Example: PCs with older DOS versions or older print servers are not DHCP-compatible and must always be configured “manually”.

All three methods can be used side by side in networks with DHCP support.

Of course, there are special cases where it makes sense to do without DHCP for address assignment. In technical applications it is often necessary to make further device-specific settings in addition to the assignment of IP address data, which are not supported by DHCP anyway.

Here, the software tools provided by the manufacturer offer more convenience than DHCP in many cases.

W&T offers the user, for example, the Wutility Tool, a tool for easy commissioning, inventory, maintenance and management of W&T devices such as Com Servers, USB Servers, Web-IO Boxes, as well as Motherboxes and pure.boxes.



Ethernet-Adresse	IP-Adresse	Produktnummer	Produktname	Version
00c03d:05a950	0.0.0.0	#57655	IP-Watcher 2x2 Digital	3.51
00c03d:affe33	10.40.21.13	#58662	Com-Server 3x Isolated	1.45
00c03d:07e6af	10.40.21.18	#58665	Com-Server++	1.47
00c03d:07de5f	10.40.21.22	#57737	Web-IO 4.0 Digital, 2xIn, 2xOut	1.31
00c03d:0665e2	10.40.21.81	#53642	USB-Server Industry Isochron	1.99
00c03d:089041	10.40.21.188	#55210	Microwall Gigabit	1.50
00c03d:089042	10.110.0.1	#55210	Microwall Gigabit	1.50
00c03d:0845d0	10.40.23.61	#57761	Web-IO 4.0 Analog 0-20mA	1.33
00c03d:06961b	10.40.26.180	#50514	pure.box 2 SD	1.55
00c03d:998877	10.40.35.20	#57734	Web-IO 4.0 Digital 12xIn, 6xRelais Out	1.33
00c03d:0715a2	10.40.41.62	#53662	USB-Server Gigabit	1.30
00c03d:0790f5	10.40.220.46	#57718	Web-Thermometer Air Quality	1.32

Of course W&T end devices, that have received their IP address via DHCP, can also be managed with Wutility.

DHCP and Router

The information exchange between end devices and DHCP servers takes place on the physical level in the form of UDP broadcasts (broadcast calls to the network). If the DHCP configuration extends over several subnets, there are two possibilities:

- The router used should work as a DHCP relay agent, i.e. support the forwarding of DHCP requests across subnets.
- A separate DHCP server should work in each subnet.

DNS – the Domain Name System

The Domain Name System is the address book of the internet. Although it is only used by the user in the background, it is one of the most important internet services.

At the IP level, the millions of participants on the internet are addressed via IP addresses. For the user, however, dealing with IP addresses would be difficult: who can remember that the W&T web thermometer can be reached at the IP address 194.77.229.26? A meaningful name, such as klima.wut.de, is much easier to remember.

Even in the early days of the internet, the need to assign symbolic names to IP addresses was taken into account: A host table was maintained on each local computer, in which the corresponding assignments were stored.

The disadvantage, however, was that only those network participants whose names were on the local list could be reached. Moreover, with the rapid growth of the internet, these local lists soon became unmanageable. The need to create a uniform name resolution system therefore arose. For this reason, the DNS standard was adopted in 1984 and has remained virtually unchanged to this day.

The principle is simple: The assignment of IP addresses and domain names is stored on so-called DNS servers and “queried” there as required. But before we go into detail here, a few remarks on the structure of domain names:

Domain names

The DNS provides for a uniform name assignment in which each individual host (participant in the network) is part of at least one higher-level “top-level domain”.

A country-specific domain name is a suitable top-level domain:

- .de for Germany
- .at for Austria
- .ch for Switzerland etc.

The domain can also be selected according to content or operator:

- .com for commercial offers
- .net for network operators
- .edu for educational institutions
- .gov is reserved for the US government
- .mil is reserved for the US military
- .org for organizations

All subordinate (sub-level) domain names can be chosen by the operator himself, but must be unique in the parent domain. For each top-level domain there is a self-governing institution from which the sub-level domains must be applied for and which thus excludes multiple allocation. DENIC (Deutsches Network Information Center; http://www.denic.de) is responsible for the de-domain in such matters.

An example: klima.wut.de is composed of:

- de for Germany as top-level domain
- wut for Wiesemann and Theis as sub-level domain
- klima for the web thermometer in the domain wut.de

The entire domain name can be a maximum of 255 characters long, with each subdomain name having a maximum of 63 characters. The individual subdomain names are separated by periods. There is no distinction between upper and lower case letters. WWW.WUT.DE leads you to the homepage of W&T in the same way as www.wut.de or www.WuT.de.

Name resolution in DNS

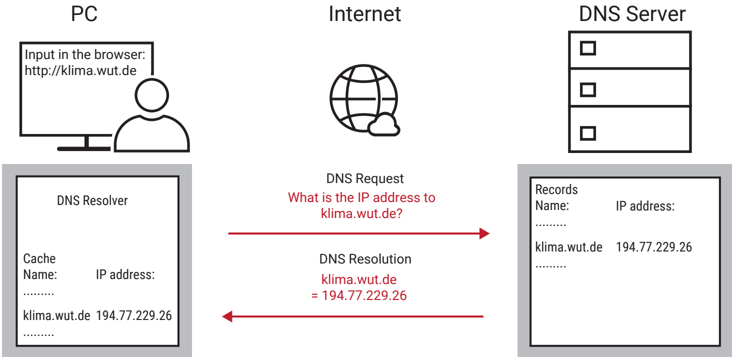
As already mentioned, DNS servers (also known as name servers) maintain lists with the assignment of domain names and IP addresses. If there were only a single DNS server in today's internet, it would be hopelessly overwhelmed by the immense number of DNS queries. For this reason, the internet is divided into zones for which one or more DNS servers are responsible.

Network participants who want to use the DNS must enter the IP address of a DNS server located in their zone in their TCP/IP stack. To be able to work even if this server fails, the usual TCP/IP stacks even require the specification of a second DNS server.

The provider or network administrator can tell you which DNS server is responsible for the respective network node.

In order to be able to resolve domain names into IP addresses, today's TCP/IP-Stacks have a resolver program. If the user specifies a domain name instead of an IP address, the resolver program starts a query to the registered DNS server. If there is no entry for the domain name searched for there, the query is forwarded to the DNS server next higher in the hierarchy. This is done until the query is either resolved or it is determined that the requested domain name does not exist.

The IP address belonging to the domain name is passed back from DNS server to DNS server and finally passed back to the resolver program. The TCP/IP stack can now address the target node in the normal way using its IP address.



The assignment of IP address and domain name is stored in a cache by the TCP/IP stack. These cache entries are dynamic: If the stored network node is not addressed for a certain time, the stack deletes the entry again. This keeps the cache lean and makes it possible to exchange the IP address belonging to a domain name if necessary.

DNS in Embedded Systems

Not all embedded systems offer the possibility to enter a domain name on the device itself.

This is not necessary anyway, because the terminal does not need to know its own name. Instead, the assignment of name and IP address is also recorded here on the DNS server. If, for example, a client is to establish a connection to an embedded system operating as a server, the client requests the IP address belonging to the name from the DNS server as usual.

However, since embedded systems work more often in “machine-to-machine connections” than in “man-to-machine connections”, direct addressing via IP address is more efficient here, since the time required for DNS resolution is eliminated.

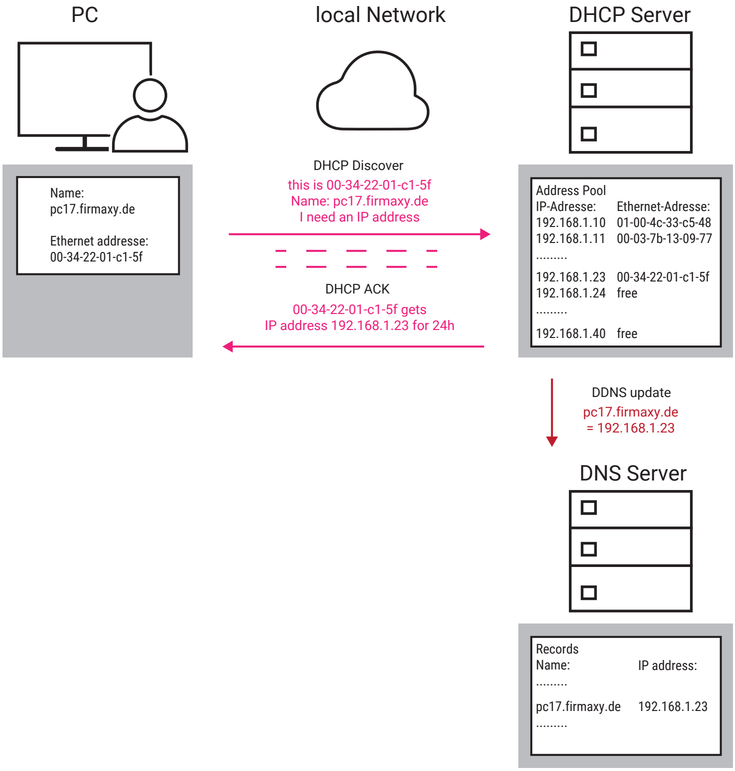
Addressing by name only makes sense for embedded systems if either only the name is known (e.g. e-mail addresses) or if a server “move” (name remains, IP address changes) must be expected (e.g. web server).

DDNS - dynamic DNS in combination with DHCP

In summary, one can say: DNS is a kind of telephone book for the network. Now DNS in its original form has the same disadvantages as a telephone book. If the telephone number of a subscriber changes after the book has been printed, the subscriber can no longer be reached with the help of this now obsolete telephone book.

The assignments in DNS servers are, of course, updated regularly and not just renewed once a year. However, if dynamic IP addresses are used, that are assigned by DHCP, DNS only makes sense if the DNS lists are constantly corrected.

The technique of automatic matching between DHCP server and DNS server is called DDNS - dynamic DNS. DDNS is not a standard TCP/IP service.



Name:
pc17.firmaxy.de

Ethernet addresse:
00-34-22-01-c1-5f

this is 00-34-22-01-c1-5f
Name: pc17.firmaxy.de
I need an IP address

00-34-22-01-c1-5f gets
IP address 192.168.1.23 for 24h

Address Pool

IP-Adresse:	Ethernet-Adresse:
192.168.1.10	01-00-4c-33-c5-48
192.168.1.11	00-03-7b-13-09-77
....	
192.168.1.23	00-34-22-01-c1-5f
192.168.1.24	free
....	
192.168.1.40	free

pc17.firmaxy.de
= 192.168.1.23

Records

Name:	IP address:
....	
pc17.firmaxy.de	192.168.1.23
....	

The method and form of synchronization between DHCP server and DNS server depend on the operating system under which the servers run.

The basic DDNS procedure for assigning an IP address via DHCP is as follows:

1. The terminal device attempts to obtain an IP address from the DHCP server. The host name of the device (here pc17.firmaxy.de) is permanently configured in the terminal device.
2. The DHCP server assigns an IP address from its address pool to the terminal device and enters the assignment to the Ethernet address in the address management.
3. In addition, the DHCP server transfers the IP address and host name of the terminal device to the DNS server.
4. The DNS server updates the name management with the new entry.

In the procedure shown, it does not matter whether the DNS server and DHCP server run on two separate computers or on the same hardware.

Since DDNS coupling must be set up by the network administrator, DDNS is only used in closed networks such as company networks.

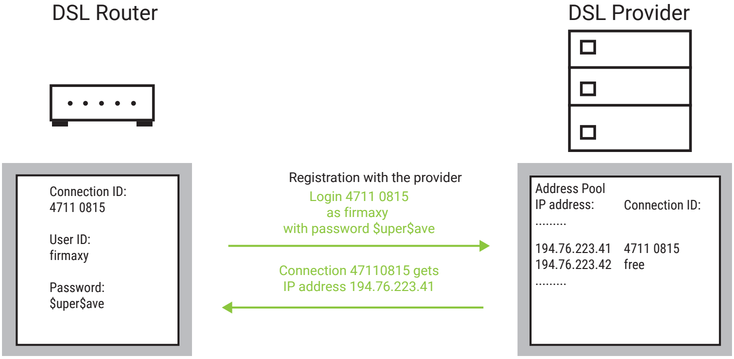
Dynamic DNS in the internet

Not only in local networks, in which the DHCP server handles the IP address allocation, dynamic IP addresses are used.

As a reminder: In networks that are interconnected - also known as WAN (Wide Area Network) - each connected terminal device must have a unique IP address. This rule applies in particular to the internet, which is by far the largest network connection.

In most cases, the connection to the internet today is made via a corresponding router. The router connects the local Ethernet network with the connection provided by the provider. For unique identification in the provider's physical network, the provider assigns a connection ID for each customer connection.

When the router is switched on, the provider assigns the connected terminal device an IP address for the duration of use, similar to DHCP. This IP address is expected to be different for each internet use.



Since most internet users only use server services (e-mail, web page retrieval, ...), i.e. they connect to these servers, this is no problem.

However, if the terminal device of the internet user (usually a PC) is also to be accessible to other internet users, the dynamic IP address is a problem, since the currently assigned IP address is only known to the provider and the terminal device connected to it.

There are two ways to avoid this problem:

1. Permanent connection to the internet

Fixed internet access with a fixed IP address is much more expensive than normal DSL access, for example. This solution is therefore only suitable for larger companies.

2. Use of Dynamic DNS

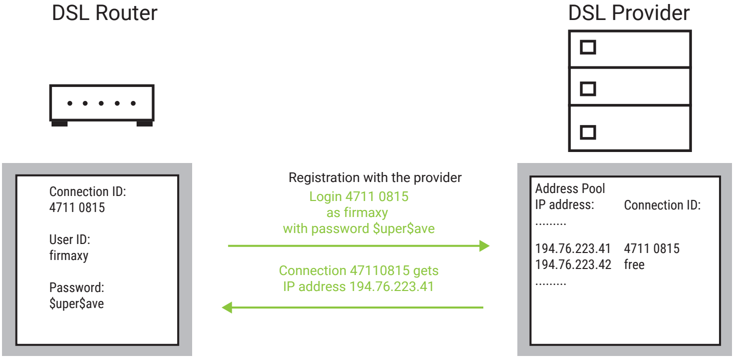
One of the first providers of dynamic DNS was the organization DynDNS. In the past, DynDNS allowed you to register a worldwide unique host name free of charge after a single registration.

Today this service is unfortunately subject to a fee.

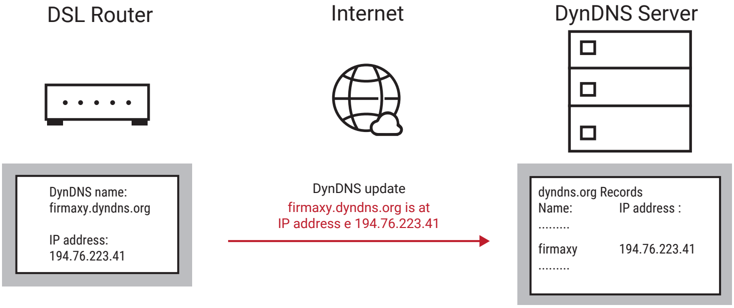
A detailed description of the procedure is available on the DynDNS web pages at http://www.dyndns.org

Address resolution using DynDNS is performed in three steps.

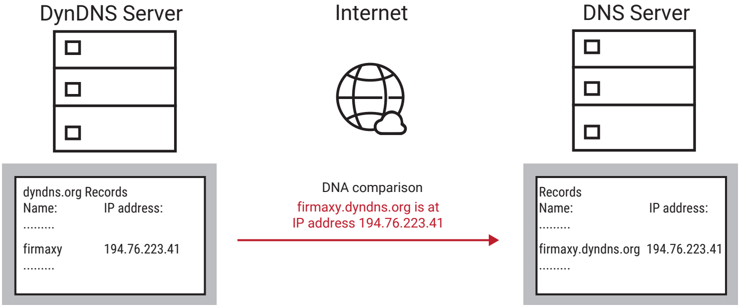
1. For example, the internet user connects to his internet provider via DSL and is assigned an IP address after successful login.



2. In contrast to DDNS, the user or his terminal device must ensure that DynDNS knows under which IP address the terminal device can be reached. The mobile device uses the DynDNS update client for this purpose. For PCs there are corresponding programs that perform this task. For other devices, special functions must be integrated. If the internet is accessed via a router, the router usually also performs the DynDNS update.



3. If a DNS server now receives a request for the DynDNS name used by the internet user and the corresponding IP address, the responsible DNS server queries the DynDNS server and compares its data.

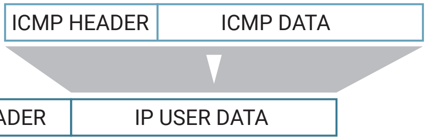


This means that the end device can be addressed worldwide under the chosen name and can therefore also offer server services.

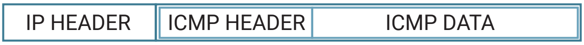
ICMP – Check reachability with Ping

The ping function is used in TCP/IP networks for diagnostic purposes. Ping can be used to check whether a particular station exists in the network and is actually addressable.

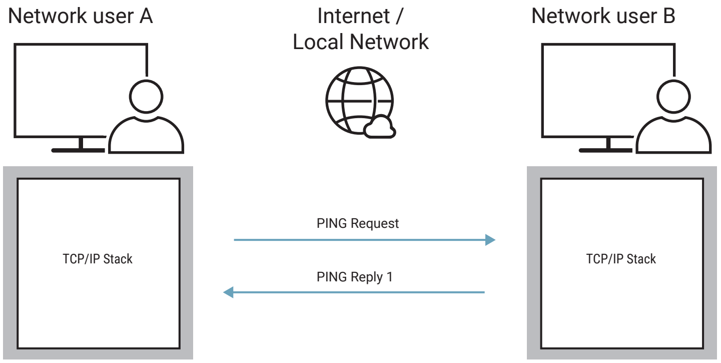
Ping works with the ICMP protocol (Internet Control Message Protocol), which is based on the IP protocol.



The package then looks like this:



If a network station sends an ICMP request by entering the ping command, the addressed station returns an ICMP reply to the sender.



The call of the command PING <IP address> in the DOS box requests the network node specified by the IP address to give feedback.

Additionally, various parameters can be specified under Windows:

-t
Repeats the ping command in a continuous loop until the user interrupts with <Ctrl> C

-n count
Repeats the ping command *count* times.

-l size
size specifies how many bytes the ICMP packet is filled with. For Com Servers in default setting this is a maximum of 560 bytes.

-w timeout
timeout specifies how long (in milliseconds) to wait for the response.

An example:

```
PING 172.16.232.49 -n 50
```

sends 50 ping commands to station 172.16.232.49. If the network station is present, the following acknowledgment appears

```
Reply from 172.16.232.49: bytes=32 time=10ms TTL=32
```

If no response is received, a corresponding message is returned:

```
Request timed out.
```

Instead of the IP address, a host name can of course also be entered. The prerequisite for this is access to a DNS server.

The ICMP packets used by Ping are defined in the internet standard RFC-792.

Application Protocols

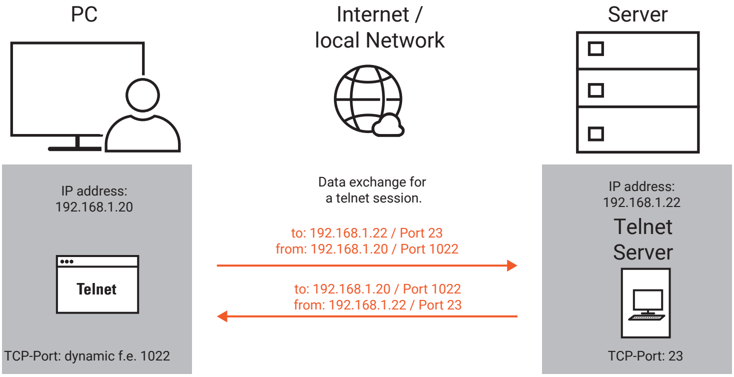
Application protocols perform a task immediately recognizable to the user or can be used directly by the user, thus creating an interface to the user.

Following on from the auxiliary protocols mentioned above, we will go into more detail about the following application protocols in this chapter:

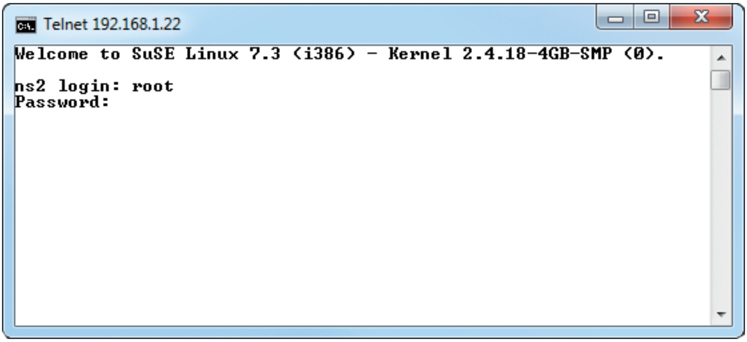
- Telnet
- FTP
- TFTP
- SNMP
- Syslog

Telnet - Terminal over Network

Simply put, Telnet is a text window or text-oriented program that allows the user to remotely control another computer (Telnet server) in the network.


PC
Internet / local Network
Server
IP address: 192.168.1.20
Telnet
TCP-Port: dynamic f.e. 1022
Data exchange for a telnet session.
to: 192.168.1.22 / Port 23
from: 192.168.1.20 / Port 1022
to: 192.168.1.20 / Port 1022
from: 192.168.1.22 / Port 23
IP address: 192.168.1.22
Telnet Server
TCP-Port: 23

A telnet session can be thought of as a DOS box, but the typed commands are executed on the remote computer.



```
Welcome to SuSE Linux 7.3 (i386) - Kernel 2.4.18-4GB-SMP (0).

ns2 login: root
Password:
```

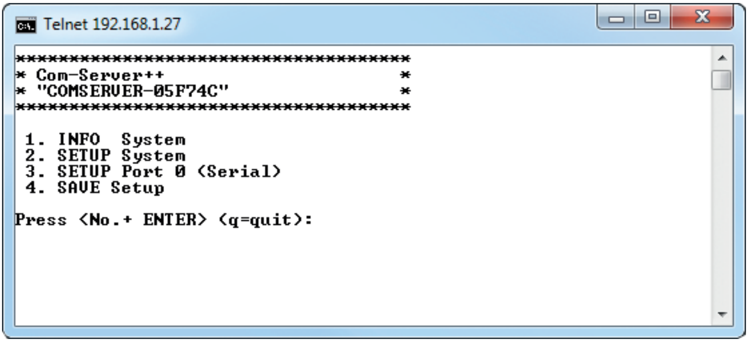
This requires several elements.

The Telnet Client

All modern operating systems today have a Telnet client program. With Windows7 or Windows10, however, the Telnet Client must first be activated. This is done in the Control Panel via Programs and Features >> Activate or deactivate Windows Features >> Telnet Client. Alternatively, third-party Telnet clients such as Putty can also be used. The Telnet client establishes a TCP connection to a Telnet server, accepts keyboard input from the user, passes it on to the Telnet server, and conversely displays the characters sent by the server on the screen.

The Telnet Server

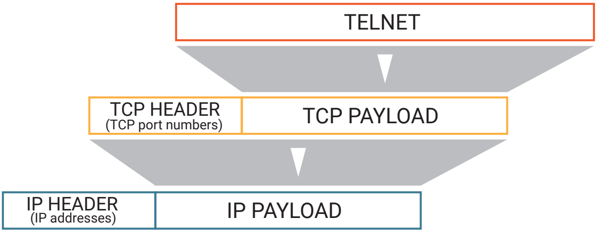
The Telnet server is active on the remote computer and gives one or possibly several users the opportunity to “log in” there. Thus, the Telnet server (often referred to as a Telnet daemon in Unix systems) is the link between network access via Telnet client and the process to be operated. In its origin, Telnet was used to provide remote access to Unix operating systems. Many embedded systems such as com servers or printer servers, switches, hubs and routers also have a Telnet server that serves as a configuration access.



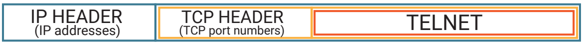
```
***************************************
* Com-Server++                        *
* "COMSERVER-05F74C"                  *
***************************************

 1. INFO  System
 2. SETUP System
 3. SETUP Port 0 (Serial)
 4. SAVE Setup

Press <No.+ ENTER> (q=quit):
```

The Telnet Protocol

Telnet is also based on TCP as the basic protocol.



The Telnet data packet then looks like this:



Unless otherwise specified by the user, port 23 is used. However, any other port can also be specified. It is important that a Telnet server is active on the selected port. The Telnet protocol essentially performs three tasks:

1. Definition of used character sets and control codes for cursor positioning

   The NVT standard "Network Virtual Terminal" is used as a common basis for client and server. NVT uses the 7bit-ASCII character set and defines which characters are displayed and which are used for control and positioning.

2. Negotiating and setting connection options

Beyond the specifications in the NTV, Telnet can make use of a large number of special functions. The Telnet protocol allows client and server to negotiate connection options. For example: whether the server should return all characters received from the client as echoes

This is done using control characters where the 8th bit is set, i.e. characters above 127 and thus outside the NTV character set.

3. Transport of characters exchanged between client and server

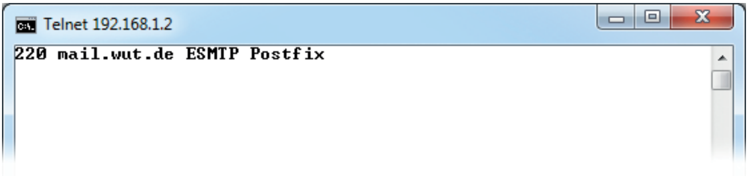
   All characters of the NTV character set entered by the user, or sent by the server, are packed 1:1 into the payload data area of a TCP packet and transported over the network.

The simplicity of the Telnet protocol and the transparency of character transmission have also made Telnet a popular diagnostic tool. It can be used to establish connections to HTTP, SMTP or POP3 servers.

For example, you can check if the SMTP server (port 25) is working by entering the following line in a dos box:

telnet <IP address of a mail server> 25

If the SMTP server is active, a welcome message is returned.



```
220 mail.wut.de ESMTP Postfix
```

By consistently typing in the SMTP protocol, one could now theoretically send e-mails via Telnet client.

FTP - File Transfer Protocol

In simple terms, FTP allows a user on a network to access the file system or hard disk of a remote computer.

One of the main applications for FTP today is the uploading of HTML pages to WWW servers, which always have FTP access for this purpose.

However, FTP can also be used to store serial data from end devices in a file on the server via embedded FTP clients, such as the W&T Com-Server.

Another field of application is data logging (cyclic storage of data records) via FTP. In this way, a W&T Web-Thermometer, for example, can write the values for temperature and humidity at specified intervals with a time stamp to a file on the FTP server.

The FTP Client

FTP works according to the client/server principle. Today, an FTP client is part of every operating system. Under Windows, for example, the FTP client is started by entering the FTP command in a Dos box.

With the OPEN command, followed by the IP address or the host name of the FTP server, the FTP connection is opened and the user must enter his login name and a password. After successful login, depending on access rights, the following file operations are possible:.

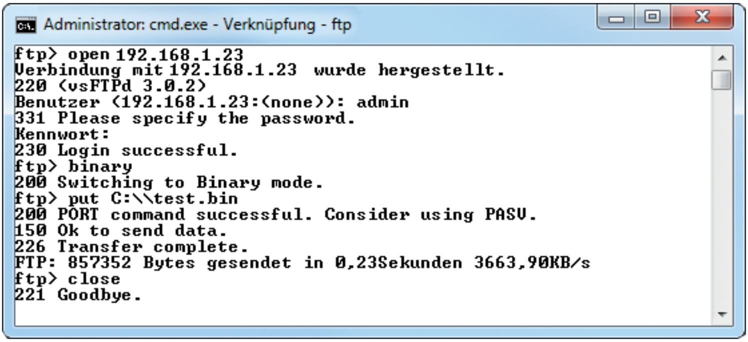
	FTP command
Saving files on the server	PUT
Load files from the server	GET
Append data to an existing file	APPEND
Delete files on the server	DELETE
Display the directory contents	DIR

A list of all supported commands can be obtained by entering a “?” after the FTP prompt. A short description of the individual commands can be obtained by entering “? command”.

An important feature of FTP is the different handling of text and binary files. To select the desired operating mode, FTP provides two additional commands:

	FTP command
for the transmission of text files	ASCII
for the transfer of binary files (e.g. executable program files)	BINARY

After entering FTP, the operation takes place in a kind of dialogue, as shown here as an example for saving the file test.bin on the server 192.168.1.23



```
ftp> open 192.168.1.23
Verbindung mit 192.168.1.23  wurde hergestellt.
220 (vsFTPd 3.0.2)
Benutzer (192.168.1.23:(none)): admin
331 Please specify the password.
Kennwort:
230 Login successful.
ftp> binary
200 Switching to Binary mode.
ftp> put C:\\test.bin
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
FTP: 857352 Bytes gesendet in 0,23Sekunden 3663,90KB/s
ftp> close
221 Goodbye.
```

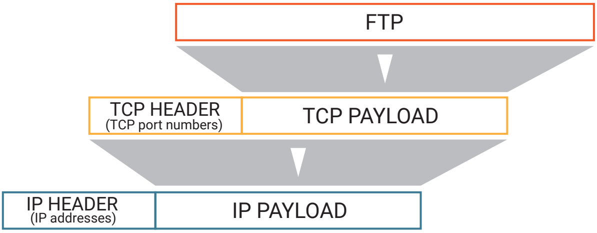
Depending on the operating system, both the operation and the commands of the FTP Client may vary.

Unix operating systems are also strictly case-sensitive.

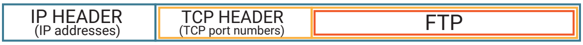
A more comfortable handling of FTP can be achieved by using purchased FTP client programs with graphical user interface.

The FTP Protocol

As a basic protocol, FTP is based on the connection-oriented and secure TCP.



The FTP data package then looks like this:

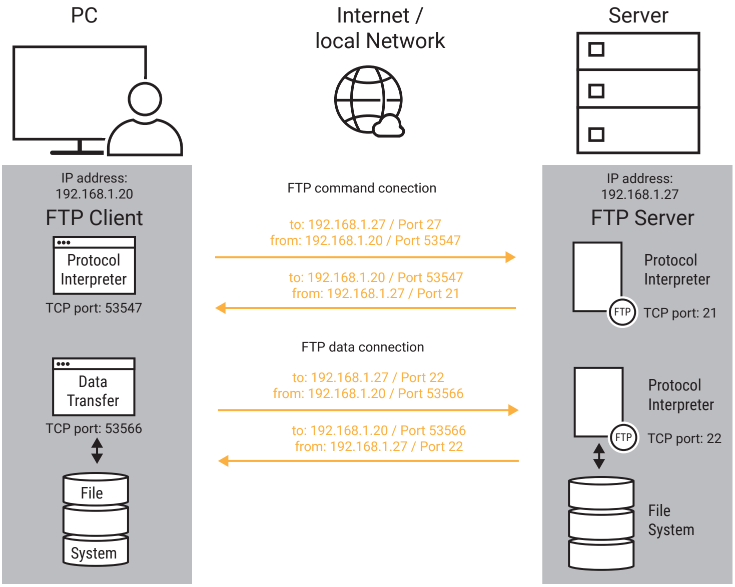


Unlike other internet services, however, FTP uses two TCP connections and thus two TCP ports:

- Port 21 as command connection
- the second port is used for file transfer. The port number used is negotiated.

The control of the file transfer between client and server is controlled by a command dialog. The protocol interpreters handle this part using the command connection. The command connection is maintained for the entire duration of the FTP session.

The actual file transfer takes place via the data connection, which is reopened by the data transfer process for each file operation. The data transfer process is the link between the network and the file system and is controlled by the protocol interpreter.



The FTP Server

An FTP server is usually only available for server operating systems and may have to be started first.

FTP servers offer two access options:

1. only registered users have access and can execute file operations, depending on the access rights recorded in a user list.
2. every user can access the server. A login either does not take place at all, or the user name “anonymous” is specified. This is called anonymous FTP.

Passive / Active FTP

FTP distinguishes between two ways of handling TCP connections.

1. Passive FTP
   The FTP Client establishes the TCP connection for both the command connection and the data connection - from a TCP point of view it acts as a client for both connections.
2. Active FTP
   The FTP Client establishes the command connection to the FTP server. The FTP Client simultaneously starts a server process for data exchange. To do this, it transmits the TCP port number on which it wants to accept the data connection to the FTP server. For the data connection, the FTP Client acts as the server and the FTP server as the client.

Active FTP and Firewalls

Cross-network FTP may cause problems if the FTP client-side network is protected by a firewall.

As a reminder, firewalls are usually configured to allow connections from the local network to the public network (internet). However, connections from the public network to the local network must be explicitly released.

Firewalls often block the incoming data connection during active FTP. In such cases passive FTP should be used.

There are firewalls that recognize active FTP and automatically release the required port for the duration of the data connection - but this is not supported by all firewalls.

TFTP - Trivial File Transfer Protocol

In addition to FTP, TFTP is another service for accessing files on a remote computer over the network.

However, TFTP is considerably “slimmer” than FTP, both in terms of the range of functions and the size of the program code.

A TFTP client is not necessarily part of the operating system.

TFTP servers are rarely used in the office environment.

TFTP is particularly suitable for use in embedded systems in which only limited storage space is available for operating system components. TFTP offers a high degree of efficiency with minimal program code.

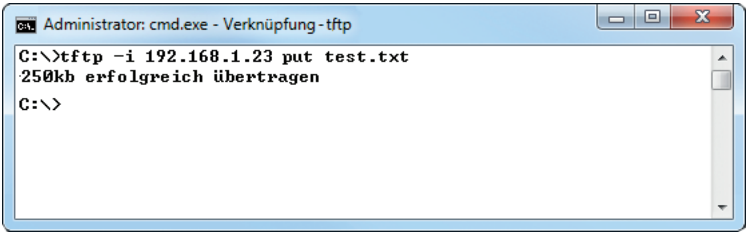
For example, TFTP is used in Com servers, printer servers and mini terminals to transfer configuration and firmware files.

TFTP provides only two file operations:

	TFTP command
Saving files on the server	PUT
Load files from the server	GET

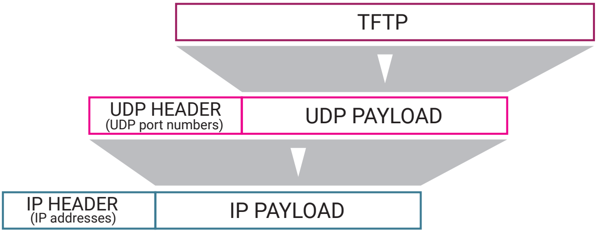
Like FTP, TFTP distinguishes between the transfer of text and binary files. If binary files are to be transferred, this is indicated by the additional parameter “-i”.

Here is a short example: The binary file “test.txt” is saved from a Windows computer to the server with the IP address 192.168.1.23.

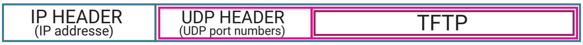


```
C:\>tftp -i 192.168.1.23 put test.txt
250kb erfolgreich übertragen
C:\>
```

Authentication, i.e. a login with password request as with FTP, is not required. In contrast to FTP, TFTP uses UDP as the basic protocol, whereby port 69 is used.



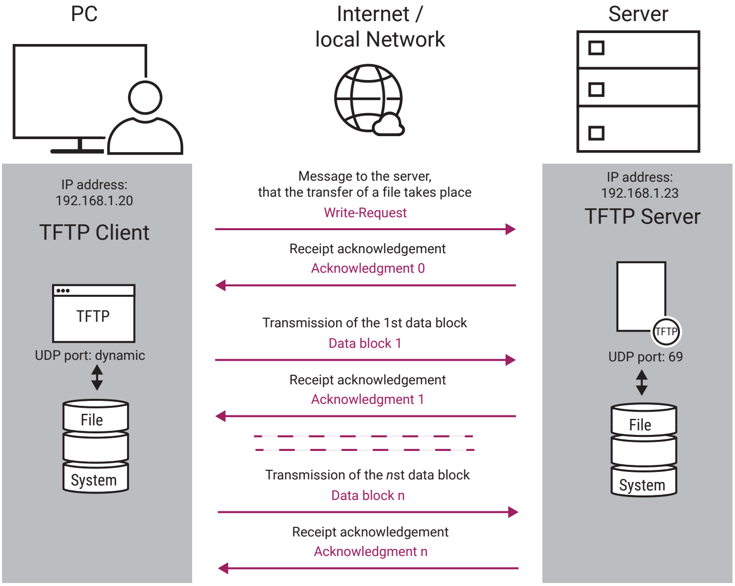
The TFTP data packet then looks like this



Remember:
UDP works without a connection. UDP packets are also called datagrams, where each packet is treated as an independent data transmission. At UDP level, received packets are not acknowledged. The sender does not receive any acknowledgment whether a sent packet has really arrived at the receiver. UDP packets do not get a sequence number. A receiver who receives several UDP packets has no way of determining whether the packets were received in the correct sequence.

For this reason, TFTP takes care of the backup of the transmitted data itself.

Files are transmitted in blocks of 512 bytes each, and the blocks are assigned a sequence number. Each received block is acknowledged by the other side. Only after receipt of the acknowledgment is the next block sent



TFTP detects whether the received data blocks are ok, but there is no error correction. If something goes wrong during transmission, e.g. the packet length is not correct or a complete packet is lost, the packet is not acknowledged by the other side. If there is no acknowledgment, the data packet is resent several times. If the acknowledgment is missing permanently, the transmission is aborted. In this case, the user or an intelligent application software can restart the process.

SNMP – Simple Network Management Protocol

Networks usually connect a large number of different end devices from different manufacturers. Each manufacturer has its own methodology for parameterizing and monitoring the devices.

For example, some manufacturers provide special management programs for their end devices, while others offer the user or administrator a web interface that can be used to monitor and configure the components in the browser.

Smaller networks can be easily set up, monitored and maintained using these tools.

In larger networks, however, with sometimes several 100 network participants, it would be very laborious to configure and monitor each device by different means.

Here SNMP provides the basis for a uniform and manageable network management.

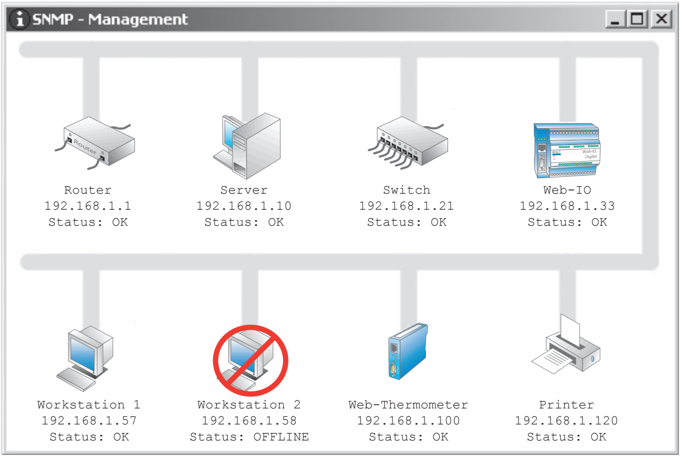
SNMP Agent

A condition for the use of SNMP is that all participating end devices have an SNMP agent. The SNMP agent is a software interface that represents the end device with all parameters important for operation. SNMP-enabled end devices are also called network nodes or nodes. Nodes can be workstation PCs, servers, switches, routers, web IOs, i.e. basically anything that can be addressed via its own IP address in the network.

SNMP Manager

In addition to the nodes, SNMP systems have at least one SNMP manager. The SNMP manager is a software application that works on a workstation or server.

While SNMP managers used to be command line controlled applications in which the nodes were managed in lists, modern SNMP systems provide the administrator with powerful visualization functions. The entire network infrastructure can be displayed in the form of plans and thus be managed very clearly.



SNMP - Management

Router
192.168.1.1
Status: OK

Server
192.168.1.10
Status: OK

Switch
192.168.1.21
Status: OK

Web-IO
192.168.1.33
Status: OK

Workstation 1
192.168.1.57
Status: OK

Workstation 2
192.168.1.58
Status: OFFLINE

Web-Thermometer
192.168.1.100
Status: OK

Printer
192.168.1.120
Status: OK

The tasks of an SNMP manager include Configuration, managing access rights, monitoring, error management and network security.

SNMP MIB

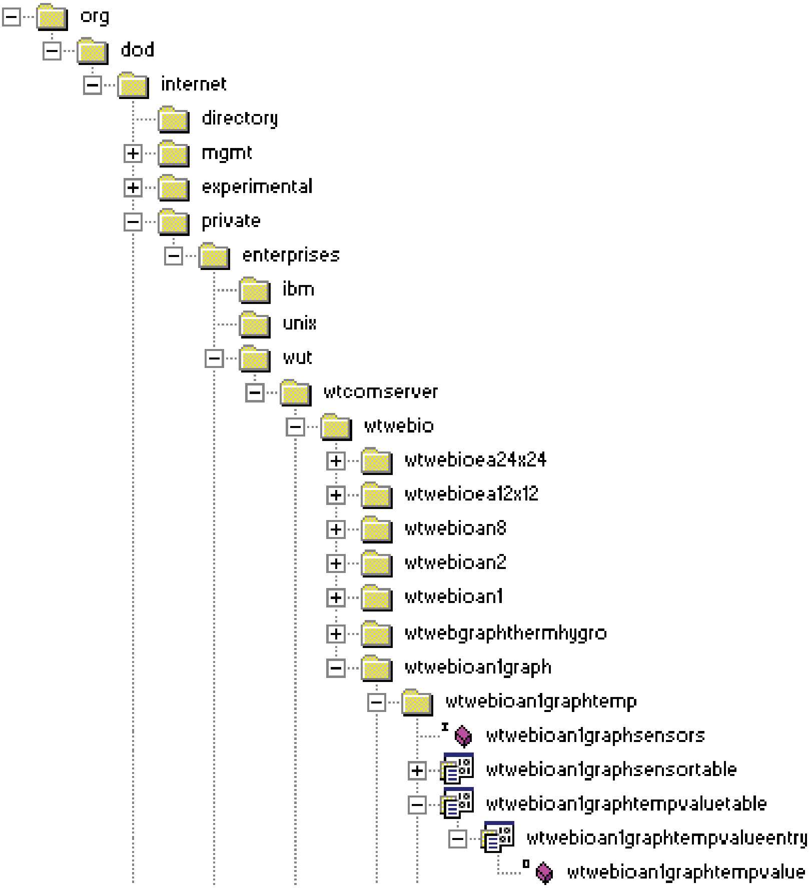
The abbreviation MIB stands for Management Information Base. Each network node has a specific MIB, i.e. a list of retrievable variables describing the properties and states of the network node.

Normally the user does not have to deal with the structure of the MIB in detail. Modern management systems have a MIB compiler that integrates the MIB data into the system and makes it available to the user in a form that is easy to handle.

In order to aid the understanding of SNMP processes, we would nevertheless like to give a rough overview of the structure here.

The MIB consists of two parts: the standard MIB, in which system variables are managed, which are needed for all nodes, and the private MIB, in which the device-specific variables are accommodated and which we will discuss in more detail here.

The data structure of the MIB has a tree-like structure, similar to the directory structure on a hard disk. The individual variables are divided into groups, subgroups, etc., just as individual files are stored on a data carrier in folders and subfolders.



```
org
  dod
    internet
      directory
      mgmt
      experimental
      private
        enterprises
          ibm
          unix
          wut
            wtcomserver
              wtwebio
                wtwebioea24x24
                wtwebioea12x12
                wtwebioan8
                wtwebioan2
                wtwebioan1
                wtwebgraphthermhygro
                wtwebioan1graph
                  wtwebioan1graphtemp
                    wtwebioan1graphsensors
                    wtwebioan1graphsensortable
                    wtwebioan1graphtempvaluetable
                      wtwebioan1graphtempvalueentry
                        wtwebioan1graphtempvalue
```

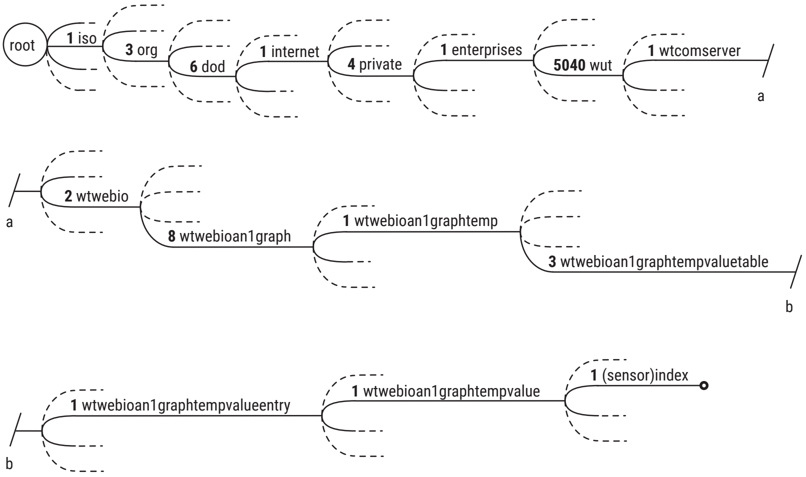
The illustration shows in the representation of a directory tree, at which point, for example in a web thermograph, the measured temperature can be retrieved via SNMP.

The MIB variables are also called objects. The MIB-OID belongs to each object of a MIB. OID stands for Object Identifier. The OID is a chain of numbers separated by dots, whereby each number indicates where to branch to in the MIB tree.

The OID for the sensor temperature of the Wiesemann & Theis Web-Thermograph looks like this

```
1.3.6.1.4.1.5040.1.2.8.1.3.1.1.1
```

Since such data chains cannot managed by the user, the OID can also be displayed as a MIB diagram:



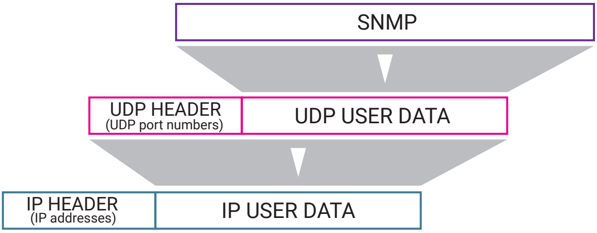
The MIB files supplied by the manufacturers of the various network nodes describe the OID structure in ASN.1 format (Abstract Syntax Notification).

ASN.1 files are readable, but decoding by the user is complicated and not intended.

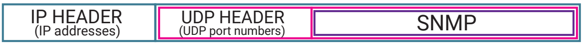
As already mentioned, SNMP management systems have an ASN.1 MIB compiler. This compiler evaluates the ASN.1 format and tells the manager which variables of a network node can be found at which position.

SNMP communication

The communication between the SNMP management system and the SNMP network node is handled via the UDP protocol.



The SNMP data packet then looks like this:



Here, the network node receives the data transmissions from the SNMP management system on port 161.

Normal communication always originates from the management system. The management system sends a GET command with the OID of the desired value to the network node. The network node then sends back a RESPONSE packet, which also contains the OID and the corresponding value. This question/answer game is also called polling.

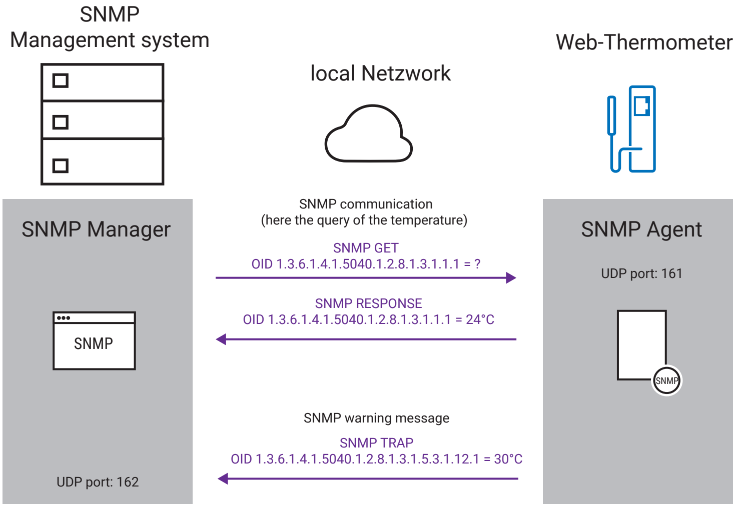
SNMP Trap

In addition to the polling initiated by the SNMP manager, SNMP gives the network nodes the possibility to send unsolicited messages to the SNMP manager.

These SNMP traps are used as status or warning messages. For example using this method a switch can report if a port loses its link, i.e. the connected end device is no longer recognized.

SNMP traps are sent to port 162.

In the case of the Web-Thermograph, alarms can be defined that send an SNMP trap when the temperature is exceeded, e.g. in the server room.



SNMP traps have their own OIDs which are located in a separate part of the MIB, even if the same value should happen to appear again in another part of the MIB.

For administrators of extensive networks with many network participants, SNMP offers all the prerequisites for handling the maintenance and monitoring of all participating devices in a uniform and clear manner.

Community Strings

The community string is a kind of password that is sent with every SNMP query. SNMP provides three different community strings for different access permissions:

Read Only Community String for read-only access

Read/Write Community String for read and modify access

Trap Community String for sending SNMP traps.

Most SNMP-enabled devices use the word “public” as the community string in all three cases ex works. However, the community strings are freely configurable.

SNMP Versions

There are now three versions of SNMP:

SNMPv1

SNMPv1 is the original version of SNMP and already includes all the features described here. One problem with SNMPv1 is the lack of security. The exchanged data passes unencrypted through the network and can be read by unauthorized persons.

SNMPv2

The main difference to SNMPv1 is that the community strings are transmitted encrypted. In addition, SNMPv2 offers the possibility to read out data summarized in a table in full with one retrieval.

SNMPv3

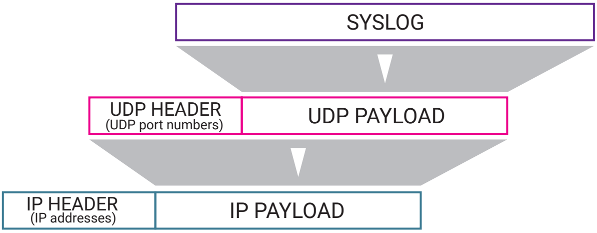
SNMPv3 enables encrypted transmission of communication data. In addition, user names and passwords are used.

For more details on encrypted data transmission, refer to the chapter Data Security/ Network Security

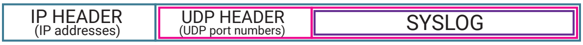
Syslog - The system logger

Similar to SNMP, Syslog is a protocol for bundling and monitoring system messages at a central location. Unlike SNMP, however, syslog is a one-way street. This means that with Syslog, network devices such as PCs, routers, switches, hubs, but also embedded devices such as Web-IO and Web-Thermographs, can send system messages to a central server; however, data transmissions from the server to the end devices are not intended.

At the network level, syslog messages are transmitted via the UDP protocol on port 514.



The SYSLOG data packet then looks like this



Syslog messages can be normal status information, warning messages and error messages.

Depending on their urgency, syslog messages are assigned priorities by the sender. In this way it can be influenced which messages are processed preferentially. Furthermore, each syslog message contains a time stamp with time and date.

The process on the server that receives and processes the syslog messages is called a syslog daemon.

Syslog originated in the Unix or Linux world, but is now also used in the Windows environment.

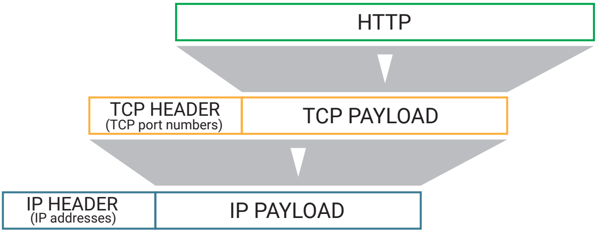
Web Protocols

The two most frequently used internet applications are the retrieval of web pages in the browser and the sending of e-mails. The protocols required for this are:

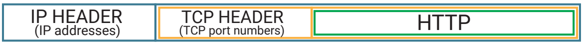
- HTTP
- SMTP
- POP3
- IMAP

HTTP/HTTPS – Hypertext Transfer Protocol

HTTP or HTTPS is the protocol that the browser uses to retrieve web pages and other content from web servers. HTTP is based on TCP as the basic protocol, whereby TCP port 80 is usually used. The S in HTTPS stands for Secure. HTTPS basically works in the same way as HTTP, but the transmission is encrypted (more on this in the chapter Data Security/Network Security). The standard TCP port for HTTPS is 443. Other ports are possible, but must be explicitly specified in the URL.

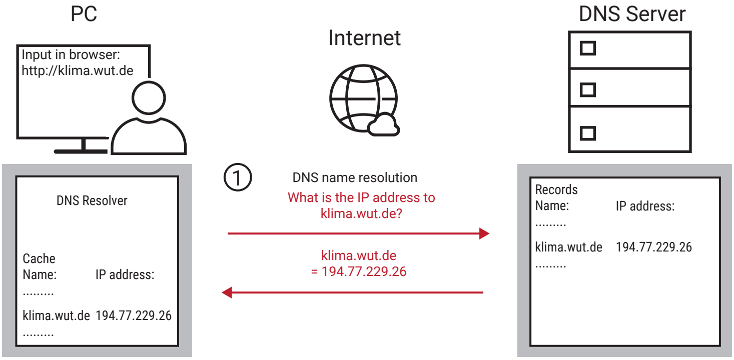


The HTTP data packet therefore has the following structure:



The request and transfer of a website is done in five steps:

1. resolving the specified host and domain name into an IP address
   The TCP/IP stack starts a DNS query to determine the IP address of the desired server.



2. establishing the TCP connection

   Reminder: The client/server principle applies to a TCP connection. With HTTP, the browser assumes the role of the client and establishes the TCP connection to the specified HTTP server.

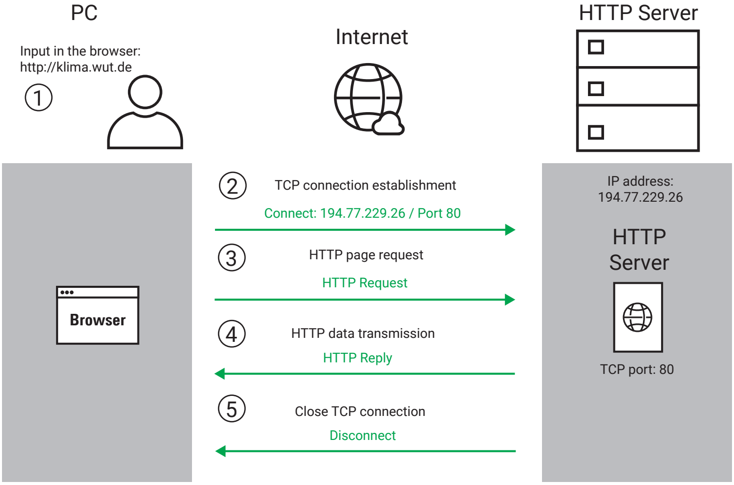
3. sending the HTTP request

   After successfully establishing the TCP connection, the browser requests the desired web page from the HTTP server. This is where the actual HTTP protocol begins: The browser sends the GET command with the required parameters to the WWW server.

4. sending the requested web page

   The HTTP server first sends an HTTP confirmation and then the website itself.

5. termination of the TCP connection by the HTTP server



A special feature of HTTP is that the TCP connection is not terminated by the client, as is usually the case, but by the server. There are two reasons for this:

- The HTTP server signals the browser in a simple way that the transfer is complete.
- HTTP servers must serve a large number of TCP connections simultaneously. Each open connection demands a certain level of performance from the server. To keep connection times as short as possible, the server simply closes the connection as soon as all requested data has been transferred.

The most important HTTP commands and parameters

As already mentioned, HTTP is also based on the client/server principle: The browser as client can control the communication by sending certain commands. Here are the two most important commands:

The GET command

By far the most commonly used command is the GET request, which initiates every call to a web page. GET requests the HTTP server to send a document or element and is therefore the most important command.

To use GET, some parameters are required; this is also called a command line (request line).

```
GET /pathname/filename http-version
```

Further parameters can be sent as new lines. These appended parameters are also called “headers”.

Host Host name (only required for HTTP1.1).
Accept specifies which file formats the browser can process
With Accept: image/gif the browser announces That it
can display images in GIF format.
Connection via this parameter, the browser can specify whether the TCP connection is kept open for reloading other elements.

A large number of other parameters are described in RFC2616. More details on https://www.w3.org/Protocols/rfc2616/rfc2616.html.

A typical GET command could look something like this:

```
GET /welcome.html http/1.1
Host: www.wut.de
Accept: image/gif
Connection: Keep-Alive
```

In response, the HTTP server sends a status line followed by a header (this time with parameters of the server). Separated by an empty line <CR LF CR LF> the requested element is transmitted:.

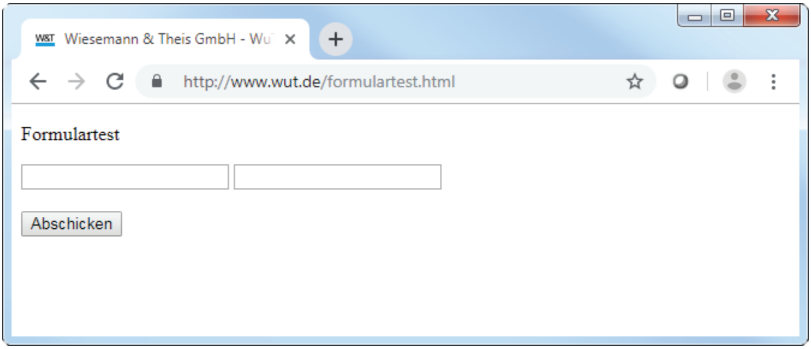
```
HTTP/1.1 200 OK                              | Status line
Date: Thu, 15 Mar 2001 11:33:41GMT           |
Server: Apache/1.3.4 (Unix) PHP/3.0.6        |
Last-Modified: Thu 15 Mar 2001 11:32:32 GMT  |
...                                          |
...                                          | Header
Keep-Alive: timeout=15                       |
Connection: Keep-Alive                       |
Content-Type: text/html                      |

<html>                                       |
  ...                                        | HTML-Site
</html>                                      |
```

The status bar includes the HTTP version supported by the server, an error code number and a comment. In the header, the server displays supported connection properties and data.

The POST command

The counterpart to GET is the POST command. POST allows the browser to transfer information to the HTTP server.

The classic use for the POST command is to transfer form entries from an HTML page. Basically, the structure of the POST request is identical to that of GET. The parameters are followed by an empty line <CR LF CR LF>, which is followed by the information to be transferred. If a POST request contains several individual pieces of information, they are separated by an "&". The filename in the first line of the POST request must be a process available on the server that can receive and process the information.


W&T Wiesemann & Theis GmbH - Wu
http://www.wut.de/formulartest.html
Abschicken

Formulartest

For this form test form, the POST request could look like this; the parameter "Referer", which has not yet been discussed, creates a reference to the originally loaded form page:

```
POST /Formularauswertung.cgi HTTP/1.1
Accept: image/gif, image/jpeg
Referer: http://172.16.232.145/formulartest.html
Host: 172.16.232.145
Connection: Keep-Alive

INPUTLINE1=test1&INPUTLINE2=test2&submit=submit
```

Tip: *Most internet providers offer so-called „CGI scripts" (programs on the HTTP server), which accept form data and forward it as e-mail to any address. This way, for example, you can give your customers the opportunity to send an order or inquiry directly from a website.*

Further commands are defined in the HTTP specification, but in practice they have almost no meaning. For the sake of making matters complete we will therefore only briefly discuss them:

- **HEAD**
  requests a web page like GET - but the HTTP server only delivers the

HTTP head back. Search engines can use HEAD to check whether a web page still exists.

- **PUT**
  is used to upload and (if it already exists) replace content to an HTTP server
- **PATCH**
  changes existing contents without replacing them completely
- **DELETE**
  deletes contents on a HTTP server
- **TRACE**
  quasi returns an echo of the sent HTTP request. This allows you to check whether an HTTP request was changed on its way to the server.
- **OPTIONS**
  returns which methods the addressed HTTP server supports.

HTTP versions

HTTP has been developed several times since the introduction of the WWW and is now available in four versions:

HTTP 0.9

HTTP 0.9 was first introduced in 1989 and has been used since, but never specified.

HTTP 1.0

Only in 1996 HTTP version 1.0 was specified by RFC 1945, which is largely identical with HTTP 0.9..

HTTP 1.1

HTTP 1.1 was introduced in 1997 (RFC 2068) and has been in use in revised form since 1999 (RCF 2616).

Probably the most fundamental change in HTTP 1.1 is that the TCP connection established for transferring the HTML document is still used for reloading other elements. HTTP 1.0 and 0.9 have established a separate TCP connection for each element. A persistent connection as in 1.1 increases the data throughput, since the times required for establishing and terminating connections are eliminated.

In order to be able to manage the internet presences of several providers on one HTTP server, an additional parameter to the GET command has been introduced with “Host”, which also transmits the host name to the server together with a

GET request (e.g. Host: http://www.wut.de). An HTTP server has only one IP address, even if it represents several host names. Thanks to this additional parameter, the HTTP server can detect which host is the destination of this connection.

HTTP 2.0

The official name is HTTP/2. In 2015 HTTP/2 was introduced as the successor of HTTP 1.1 with the following extensions:

- several requests can be combined
- faster transmission through data compression
- binary data are supported in addition to text
- the server can send data on its own initiative (without request)
- the transmission of certain contents can be prioritized

This makes HTTP/2 significantly faster and more flexible in data transfer than its predecessors.

All current browsers support HTTP/2 by default, but can also work also with servers that use HTTP 0.9, HTTP 1.0, or HTTP 1.1.

Browser Cache and Proxy Server

As already mentioned, HTTP is one of the most used protocols. Therefore, HTTP usually accounts for a high percentage of the data transmitted over the internet.

Especially with web pages and their contents such as pictures, for example, it is the case that by repeatedly calling up the same web page, the same contents are called up again and again.

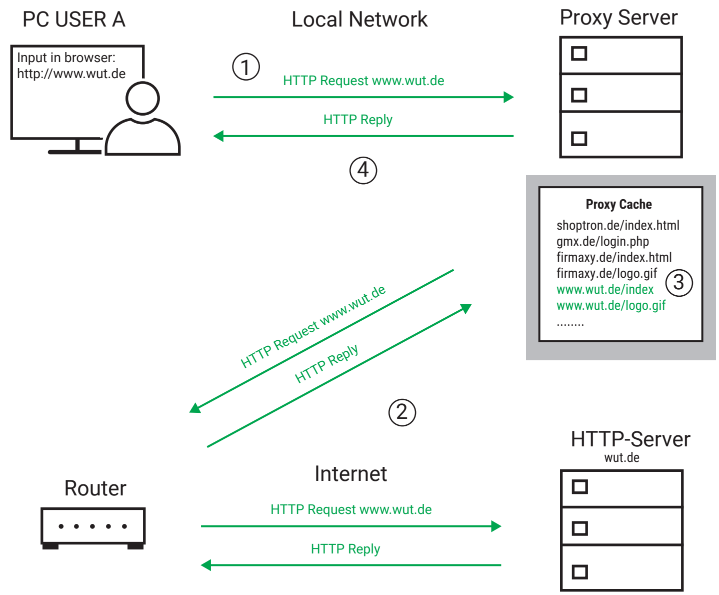
This means that the same content is loaded unnecessarily multiple times from the internet.

In order to limit double loading of data, the common browsers offer a so-called cache - a buffer in which loaded contents are temporarily stored.

Proxy servers are also often used in the larger networks of companies, universities and other institutions. HTTP requests are redirected to the proxy server, which, similar to the browser cache, has a memory in which the retrieved content is temporarily stored.

While the browser cache only stores the content retrieved by one user, the proxy server holds the content retrieved by all users.

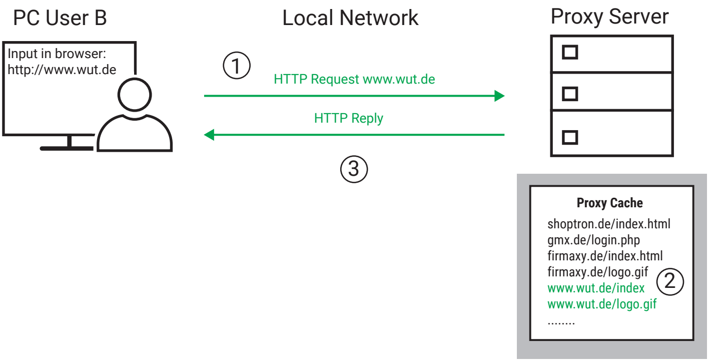
Example:
User A calls up the website of Wiesemann & Theis



1. the HTTP request http://www.wut.de is sent to the proxy server
2. the proxy server determines that it does not yet know the content of www.wut.de and forwards the request to the HTTP server where www.wut.de is hosted. The HTTP server sends the requested content to the proxy.
3. The proxy server stores the contents of www.wut.de in its cache.

4. The proxy server answers the HTTP request of the browser with the contents of the HTTP server.

Later, user B also calls up the website of Wiesemann & Theis



1. The HTTP request http://www.wut.de is sent to the proxy server
2. The proxy server determines that it already has the contents of www.wut.de in its cache.
3. The proxy server answers the HTTP request of the browser with the contents from its cache.

In this case, no data will be transmitted via the internet.

Thus, the browser cache and the use of proxy servers can reduce the volume of data via internet access.

Retrieving current data

However, there are also website calls where it is not desired to get possibly outdated data from a cache. As far as the browser cache is concerned, the user can use the key combination <Control + F5> to instruct the browser to retrieve the desired web page from the server at the current time.

In the case of websites that make it absolutely necessary to provide current data each time they are called up, this can be specified via corresponding header entries, i.e. specifications in the header of a website.

E-Mail

The ability to send electronic mail from one end of the world to the other in a few seconds is certainly one of the main reasons for the rapid spread of the internet.

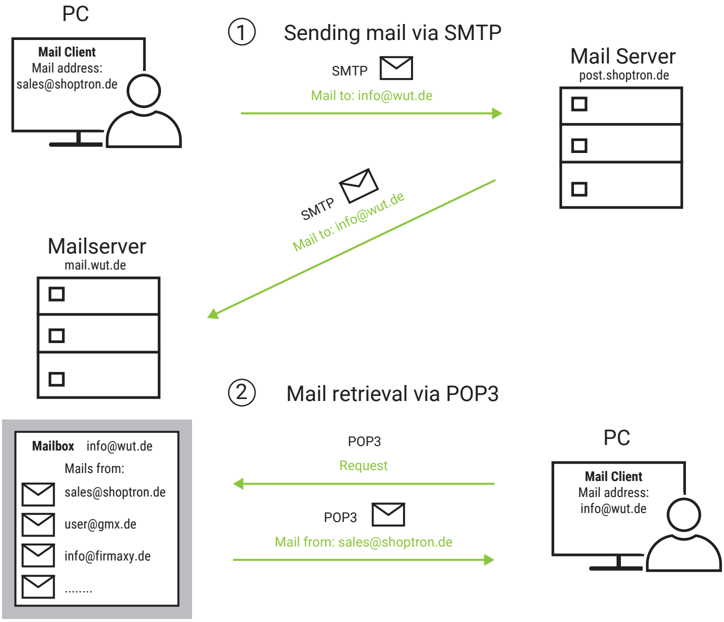
Unlike most other applications on the internet, sending e-mails is a service where there is no direct connection between sender and recipient. This sounds confusing at first, but it makes sense, because otherwise the exchange of e-mail would only be possible if sender and recipient were active on the net at the same time.

In order to guarantee temporal independence, the e-mail recipient needs a mailbox on a mail server, in which incoming messages are first stored.

An e-mail address is always composed of the mailbox name and the target domain; the “@” (English “at”) separates these two components. An example: info@wut.de is the info mailbox on the mail server of Wiesemann & Theis.

The path of an e-mail from the sender to the recipient consists of two subsections, in which the transport is regulated by different protocols:

1. The SMTP protocol is used from the sender’s computer to the recipient’s mailbox
2. The POP3 protocol is used from the recipient’s mailbox to the recipient’s computer



Structure of an E-Mail

An e-mail consists of the message header and the actual message. This header can be compared to an envelope, which contains fields for sender, recipient, date, subject and some more information.

Here is an overview of the most important fields:

The following five fields form a minimum header and must be included in any case.

Field	Function
FROM	E-Mail address of the sender
TO	E-Mail address of the recipient
DATE	date and time Note: the time can be entered arbitrarily and is usually the local time of the sender
SUBJECT	Text of the subject line

Field	Function
RECEIVED	The field RECEIVED is a special feature because it is not created when the e-mail is created. Each mail router on the path of the e-mail inserts a RECEIVED field and thus leaves a „transit stamp“ with date and time.

The use of the fields listed below is optional.

Field	Function
SENDER	E-Mail address of the sender (usually identical with entry under FROM)
REPLY-TO	E-Mail address to which the recipient should reply. Important if e-mails are sent automatically by an embedded system such as the W&T Web-IO. In this case, the e-mail address of the administrator could be entered as the reply address.
CC	E-Mail address of another recipient who receives a carbon copy (CC) of the message.
BCC	E-Mail address of another recipient, but which remains invisible for all other recipients (BCC = „Blind Carbon Copy“)
MESSAGE-ID	Unique identification of an e-mail, which is arbitrarily assigned by the mail software.
X-“MY FIELD“	You can create your own fields by prefixing them with „X-“.

For some fields, a RESENT variant is possible, which comes into effect if the e-mail is forwarded by the original recipient.

The formal structure of the message header and fields must satisfy the following conventions:

- The field name is followed by a colon; the respective parameter follows.
- Each field is on a separate line ending with <CR LF> (Carriage Return Line Feed; hex 0D 0A).
- Message header and body are separated by an additional blank line <CR LF>.
- The message body itself contains only the text to be transmitted or further inserted files. The end of the message is indicated by <CR LF . CR LF> (hex 0D 0A 2E 0D 0A).

- Both header and message body consist exclusively of 7-bit ASCII characters. Therefore, all control information can also be transmitted as plain text.

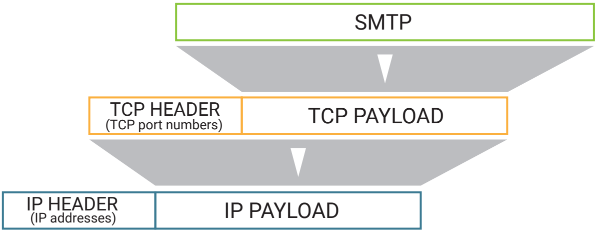
MIME – Multipurpose Internet Mail Extensions

In order to be able to send binary data (8-bit format) via e-mail, these are encoded into 7-bit format according to the “MIME standard” before being integrated into the message body and decoded again at the recipient’s end. Since the processing of binary data is automatically taken over by today’s mail programs, we will not explain the “MIME coding” in detail here.

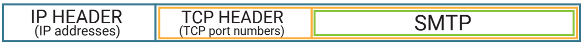
SMTP – Simple Mail Transfer Protocol

SMTP controls the sending of e-mails from the mail client to the mail server (SMTP server). The mail client can either be the original sender or a mail router on the way. Mail routers are used if the e-mail is forwarded over several domains on its way. The term MTA (mail transfer agent) is often used for mail routers.

A separate TCP connection is established for each part of an e-mail. SMTP is based on this TCP connection, using TCP port 25.



The structure of the SMTP data packet is as follows:



SMTP provides some commands (e.g. specification of the sender, specification of the recipient, ...). Each SMTP command is acknowledged individually by the SMTP server. The actual e-mail is sent complete with header and body and only then acknowledged by the SMTP server. If there are no further e-mails to be sent, the TCP connection is also terminated.

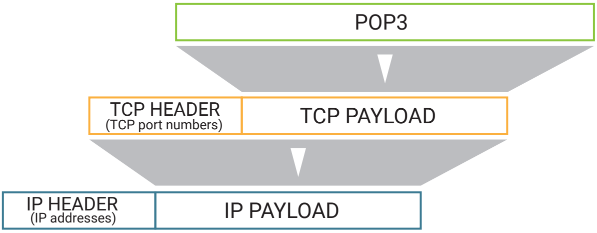
Once the e-mail has reached the destination mail server, it is stored in the recipient's mailbox and remains there until it is picked up by the recipient.

POP3 – Post Office Protocol Version 3

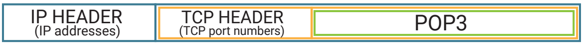
To retrieve incoming e-mails from the mailbox on the mail server, the POP3 protocol is used in most cases. The recipient is not informed about incoming e-mails. He has to check his mailbox for incoming e-mails himself and can pick them up at any time.

Most of the mail programs in use today automatically check the user's mailbox for incoming mail when they are started. Many e-mail programs also offer the option of specifying an interval at which the mailbox is checked cyclically. Typical users who are "offline" most of the day only receive their e-mails if they have logged on to their provider anyway. However, cyclical checking is certainly useful for computers with permanent internet access: here the user is constantly online and receives his e-mails with only a slight delay - virtually in real time.

The POP3 protocol is also based on a TCP connection and is nothing more than a plain text dialog, i.e. an exchange of readable commands.



The structure of the POP3 data packet therefore looks as follows:



POP3 uses the TCP port number 110 and, as with SMTP, the dialog starts with a login. With POP3, however, the recipient must log in in two steps: with user name and with password. After successful login, POP3 provides some commands to list, retrieve or delete incoming messages.

Today, the user is confronted with SMTP and POP3 only to a limited extent: He only has to specify the name of the POP3 and SMTP server when setting up the mail software - the handling of the protocols themselves is done invisibly in the background by the mail program.

For the sake of completing the picture, it should be mentioned that in addition to the POP3 protocol, there are also the POP2 and POP1 protocols (both precursors of POP3), which were also developed for fetching email. However, these protocols were not able to establish themselves in practice, or were replaced by POP3.

IMAP - Internet Message Access Protocol

Just like POP3, IMAP is based on TCP as the basic protocol and is used to transport received e-mails into the client application. Unlike POP3, IMAP leaves received emails on the server and fetches only a copy of the email into the client application for viewing.

For the user, this has the advantage that an e-mail account can be used by different end devices such as a PC, notebook, smartphone or tablet and all devices see the same reception status.

Another innovation of IMAP is the possibility to store and manage received e-mails in folders on the mail server.

E-Mail via SMTP with authentication

SMTP in its original form does not require the user who wants to send e-mails to authenticate himself in any way, i.e. to prove his authorization.

This means that anyone who has access to the network in which the SMTP server is placed can send e-mails from there.

In the age of internet, spam (unsolicited advertising e-mail) and computer viruses, this is of course an unacceptable state of affairs.

Therefore, authentication procedures have been developed that only allow the authorized user to send e-mails via the server.

We would like to briefly introduce the two most common procedures here.

SMTP after POP3

This method is very simple. Only those users who have a POP3 mailbox on the mail server are authorized to send e-mails via this server.

Before sending e-mails is allowed, a login to the POP3 mailbox must be made.

The advantage of this method is that every normal mail program first scans the POP3 mailbox for new incoming e-mails after it has been started and automatically creates the conditions for sending e-mails via the associated POP3 login.

The user therefore does not need to make any special configuration of his mail program.

An exception is made for embedded end devices such as Web-IOs or Web-Thermometers. Here, “SMTP after POP3” must be set as SMTP authentication, because these devices do not receive e-mails and therefore do not automatically access the POP3 mailbox.

ESMTP - Extended SMTP

If ESMTP is used, authentication takes place within the SMTP communication. E-mails can be sent independently of POP3 access.

Once the TCP connection to the SMTP server has been established, the server first asks for a user name and the corresponding password.

Only when both have been correctly transferred can e-mails be sent.

For the operation of embedded devices this method has the advantage that only a TCP connection is required to send e-mails.

Normal mail programs must be specially configured for ESMTP operation.

Encrypted e-mail transmission with SSL/TLS

Particularly in the case of public e-mail providers, it is now expected that data exchange when sending e-mails will be encrypted.

For the user this does not really change much. The encryption is done by the mail client and mail server under the hood, so to speak, and is not even noticeable to the user.

The SSL/TLS procedure used for this is explained in more detail in the chapter Data Security/Network Security.

However, the user must take into account that other TCP ports are used for encrypted e-mail transmission, which may vary depending on the provider:

- SSL/TLS SMTP: TCP port 465 (or 587)
- SSL/TLS POP3: TCP port 995
- SSL/TLS IMAP: TCP port 993

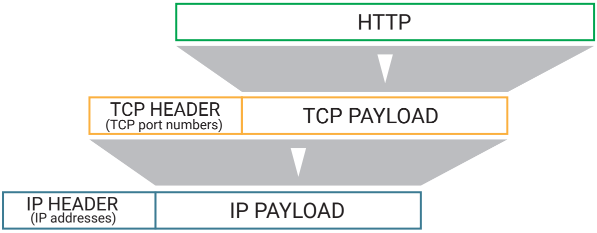
Send and receive e-mail via HTTP

With the increasing use of e-mail, more and more freemail providers, who provide mailboxes on their mail servers free of charge, have established. This service, which anyone can use, is usually financed by advertising.

In order to create space for the display of advertising, most freemail providers give the user the opportunity to send and retrieve e-mails conveniently via HTTP in the browser, which is of course enriched by advertising banners. For this purpose the user is provided with corresponding HTML forms.

In order to enable e-mail processing via HTTP, the freemail provider must operate a special mail server combination that works as a web server on the user side and as an SMTP server on the other side. The path of an e-mail looks like this:

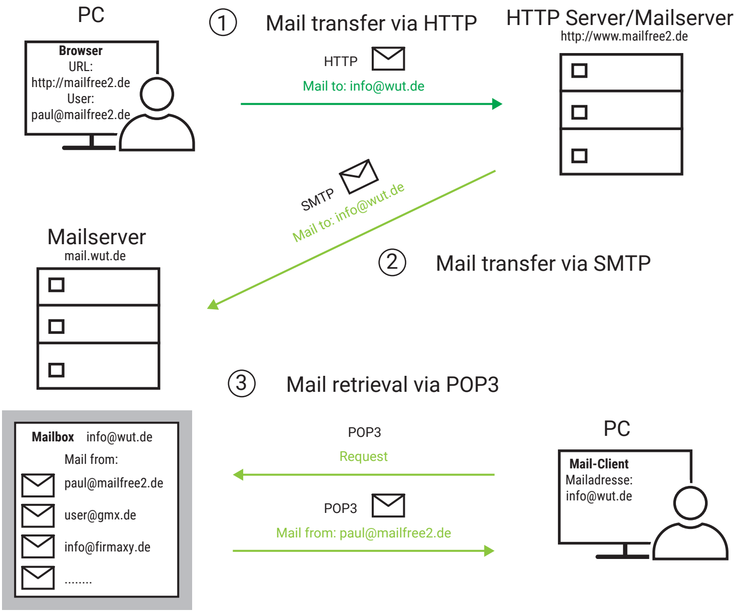
1. The HTTP protocol is used between the sender’s computer and the server of the freemail provider As with other HTTP applications, the TCP port number 80 is used.



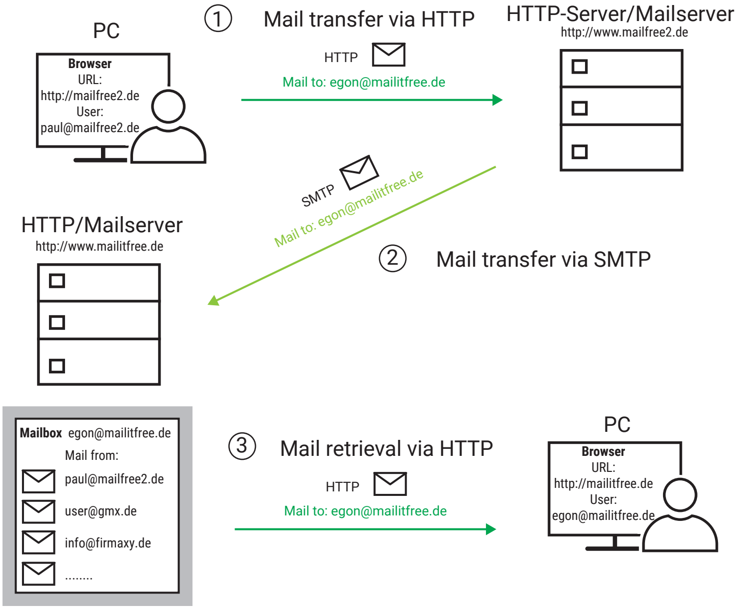
2. nothing changes between the mail servers themselves. They communicate with each other using the SMTP protocol.

3. two different variants can be used between the target mail server and the recipient's computer:

   If the recipient has a standard mail account, incoming mails are fetched via POP3.



If the recipient also uses the services of a freemail provider, then HTTP is also used.



If you prefer to send your e-mail via SMTP and POP3, you should make sure that access via an SMTP or POP3 server is available when choosing a freemail provider.

E-Mail and DNS

When sending e-mails, IP addresses are also used at the IP level. In principle, name resolution for e-mail addresses works in the same way as for normal network nodes. Of course, the address of the e-mail recipient itself is not resolved, but only that of the mail server on which the recipient has his or her mailbox.

Remember: To resolve names into addresses, the TCP/IP stack uses a resolver program that makes a corresponding request to the DNS server.

In this case, however, the host name of the target mail server is not known. The only thing known is the target domain, which is the one that appears after the @ sign in the e-mail address. In order to be able to resolve DNS queries for mail servers, DNS servers have special data records in which the mail servers belonging to a domain are listed together with the corresponding IP addresses.

The resolver program therefore only specifies the target domain name in the query and also informs you that the network node you are looking for is a mail server. The

DNS server determines the IP address that is being searched for and passes is back to the resolver program.

The mailbox name itself is not required for the DNS query. It is only evaluated when the message arrives at the target mail server so that it can be stored in the correct mailbox.

Industrial Protocols up to IoT

There has been much hype surrounding the topic of industry 4.0, but even before this efforts have been made to create standardized communication possibilities for industrial use.

In the past, fieldbus systems, i.e. serial connections between the components involved, were often used. Various standards have established themselves side by side, which differ not only in protocol and transmission speed. The physical transmission and the mechanical connection options used also vary greatly.

The industrial protocols presented in the following may differ at protocol level, but all use TCP/IP Ethernet as the physical transport medium.

This means that there is a common standard that offers many advantages:

- existing infrastructure can be used
- different industry protocols can be used side by side in the same network
- uniform transmission technology and connectors
- Communication across locations possible
- arbitrarily expandable

IoT and Industry 4.0

Both terms are currently on everyone's lips, but are interpreted quite differently in some cases. Therefore, we will limit ourselves to a short description here and then concentrate on the protocols used.

IoT - Internet of Things

The rapid expansion of the internet is certainly one of the greatest achievements of our time. Until recently, the main use of the internet was that a user, i.e. a person on a PC or smartphone, could call up web pages or send e-mails in a browser. In the meantime further services such as Twitter, WhatsApp, various smartphone apps and the streaming of videos have all been added.

But in all cases a human has been involved in this exchange of information.

The Internet of Things is not just about enabling PCs, tablets and smartphones to access the internet. Ultimately, devices with a wide variety of functions are supposed to communicate with each other via the internet.

The fields of application range from simple temperature monitoring to smart home technology and autonomous driving of cars. In industry, too, the internet is increasingly becoming a communication channel for the exchange of machine and production data.

Industry 4.0

The term Industry 4.0 seems a bit strange, because you had never heard of Industry 1.0, 2.0 or 3.0 before. Ultimately, the term was brought into play by politicians to emphasize the importance of digitization in the industrial environment.

In the process, previous generations were also assigned to specific epochs.

- **Industry 1.0**
  From 1800 onwards, machines were increasingly used for mass production. With the invention of the steam engine, machines such as looms could also be operated independently of the water power used until then. More and more factories were built.

- **Industry 2.0**
  With the introduction of electricity at the end of the 19th century, the drives for machines became smaller and lighter. Conveyor belts and piecework found their way into factories

- **Industry 3.0**
  From the 1970s onwards, the first computers were used in production plants. PLCs (programmable logic controllers) monitor, control and automate production processes.

- **Industry 4.0**
  The goal of Industry 4.0, which is being considered by politicians, is individual and self-sufficient mass production. The idea is that all components involved in a manufacturing process should automatically help to shape the process. The goal is “batch size 1”, which means that modern production should be flexible enough to allow individual pieces to be produced between mass production. And this without external intervention and without machine conversion.

In particular, the border between Industry 3.0 and Industry 4.0 is fluid. With ever more powerful hardware, the need for data exchange is constantly growing. As far as data communication is concerned, there has long been a uniform use of TCP/IP both in the local area within company networks and in the remote area using the public internet. New in this context is the increasing demand for security. You can read more about this in the chapter Data Security/Network Security.

It is now an established fact that the increasing digitalization in factory halls is already in full swing and can no longer be stopped.

We will briefly introduce the most important established standards of industrial TCP/IP-based data communication in the following.

Message formats

In the previous chapters we have already got to know some protocols. All protocols have one thing in common: There is basic address information and the actual data to be transmitted. Optionally, there are checksums or other information to secure the data.

The form in which the transported data is transmitted depends on the application and in the case of most protocols it is predefined.

There are two basic data formats:

- Binary data
- Message text

When which variant comes into effect depends on many factors.

Binary data

Remember: Data is always a certain number of bytes.

Which byte serves which purpose in which position is determined either by a standardized protocol or the application. Behind one or more bytes is a value, an array of values, a character string or even a function call.

Individual values can be transmitted in a data transmission. Often, however, data structures are used in which it is predetermined which value will be stored at which position in the transmitted byte chain.

Here as example data of a Modbus function call. The function code, for example, is always stored in the 8th byte:

Transaction ID		Protocol ID		Length		Unit ID	Funct. Code	Start Address		Number of Registers	
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12
0x02	0xA7	0x00	0x02	0x00	0x06	0x01	0x01	0x10	0x20	0x00	0x02

Another common method for binary data structure is TLV, which stands for Type Length Value. Multiple contents of any size can be transmitted consecutively in one data transmission.

The following sequence applies for each content:

1. type - what kind of content is it?
   type determination determined by the application
2. length - how many bytes does the content contain?
3. value - bytes of the value or content

If there is more content following behind such a sequence, a new sequence will start over for this content.

Here is a simple example:

Type	Length	Value		Type	Length	Value			
Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10
0x10	0x02	0xB3	0x17	0x55	0x04	0x08	0x15	0x47	0x11

The transmitted bytes contain two values: a 16-bit value (2 bytes) and a 32-bit value (4 bytes).

The advantage of binary data transmission is the very compact structure of the data.

Data as text

Especially with web-based applications, data of all kinds is sent in text form. Text means that the information is transmitted as a string of characters readable by humans. Each character occupies one byte.

In the past, the coding was based on the ASCII standard. The assignment of which character corresponds to which numerical value is defined in the ASCII table (ASCII = American Standard Code for Information Interchange).

The special feature in the past was that only 7 of the 8 available bits of a byte were used, which limits the number of characters that can be used to 128 readable characters.

Newer standards such as UTF8 overcome this limitation and allow for special characters even two bytes for one character.

In addition to freely formulated text content, standardized text formats have established themselves in web and industrial protocols:

- XML
- JSON

Both formats will be briefly explained here.

XML- Extensible Markup Language

XML is a so-called markup language. The actual user data is embedded in tags. The tags are names of the respective values or contents. Each tag begins with an opening angle bracket and ends with a closing one.

Each XML construct begins with a start tag, in which at least the XML version is specified. Additional parameters, such as the character encoding used, are also possible:

```
<?xmlversion="1.0" encoding="UTF-8"?>
```

The start tag is followed by the other content embedded in tags. All contents except the start tag have an opening and a closing tag of the same name. However, at the closing tag the naming starts with a slash (“/”).

Example:

```
<tag>irgendetwas</tag>
```

XML also allows hierarchically structured nested tags. Here as an example the sensor values of a W&T Web-Thermo-Hygrobarometer:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<webio>
  <iostate>
    <sensor>
      <name>Temperature</name>
      <number>0</number>
      <unit>°C</unit>
      <value>23.900000</value>
    </sensor>
    <sensor>
      <name>rel. humidity</name>
      <number>1</number>
      <unit>%</unit>
      <value>36</value>
    </sensor>
    <sensor>
      <name>Peassure</name>
      <number>2</number>
      <unit>hPa</unit>
      <value>992</value>
    </sensor>
  </iostate>
</webio>
```

The indentations are not obligatory with XML, but they are common, since the readability is increased considerably.

The advantage of using XML as a transmission format is that both man and machine or an evaluating program can read the contents easily.

The disadvantage is the very high gross data volume for little content.

JSON - JavaScript Object Notation

The syntax, i.e. the structure of JSON, is based on a subset of the JavaScript syntax.

JSON uses pairs of name and value/content to encode the data.

Example: “content” : “something”

JSON also allows a hierarchically constructed nested structure. Here as an example, once again the sensor values of a W&T Web-Thermo-Hygrobarometer:

```
{
  "iostate":
  {
    "sensor":
    [
      {
        "name":   "Temperature",
        "number": 0,
        "unit":   "°C",
        "value":  24.1
      },
      {
        "name":   "rel. humidity",
        "number":1,
        "unit":   "%",
        "value":  35.9
      },
      {
        "name":   "Peassure",
        "number":2,
        "unit":   "hPa",
        "value":  991.8
      }
    ]
  }
}
```

Both names and values are embedded in quotation marks at the top. An exception are numerical values - here you can do without the quotation marks.

Name/value pairs are separated by commas.

Name/value pairs that belong together must be grouped together using curly brackets.

Groups that belong together can form an array and are separated by commas and are enclosed in square brackets.

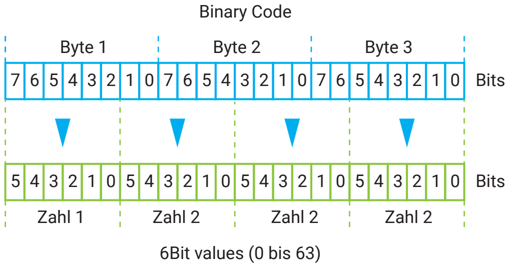
A detailed description of the JSON format is available at https://www.json.org.

JSON is much more compact in terms of data volume than XML, but still easy to read by humans and machines.

Base64 encoding

Base64 is a method that encodes or decodes binary data into a chain of readable ASCII characters. In this way, binary contents can also be transported with text-based transmission formats.

The procedure is quite simple. Three bytes of the binary code are transferred bit by bit to four 6-bit numbers.



Each of the four numbers is assigned the character corresponding to the value according to the following table. In this way, three binary bytes are replaced by four chars, i.e. readable characters.

Value		Char	Value		Char	Value		Char	Value		Char
dec.	hex.		dec.	hex.		dec.	hex.		dec.	hex.	
0	00	A	16	10	Q	32	20	g	48	30	w
1	01	B	17	11	R	33	21	h	49	31	x
2	02	C	18	12	S	34	22	i	50	32	y
3	03	D	19	13	T	35	23	j	51	33	z
4	04	E	20	14	U	36	24	k	52	34	0
5	05	F	21	15	V	37	25	l	53	35	1
6	06	G	22	16	W	38	26	m	54	36	2
7	07	H	23	17	X	39	27	n	55	37	3
8	08	I	24	18	Y	40	28	o	56	38	4
9	09	J	25	19	Z	41	29	p	57	39	5
10	0A	K	26	1A	a	42	2A	q	58	3A	6
11	0B	L	27	1B	b	43	2B	r	59	3B	7
12	0C	M	28	1C	c	44	2C	s	60	3C	8
13	0D	N	29	1D	d	45	2D	t	61	3D	9
14	0E	O	30	1E	e	46	2E	u	62	3E	+
15	0F	P	31	1F	f	47	2F	v	63	3F	/

This process is repeated until all of the binary bytes are encoded. If individual bytes remain at the end, fill bytes are added to encode the last three bytes. Fill bytes have the value 0.

In order to be able to sort out the fill bytes again during the subsequent decoding, i.e. the recovery of the original binary bytes, a “=” character is appended to the encoded character string for each fill byte.

Base64 encoding is most commonly used for web-based applications and e-mail.

Modbus-TCP

Originally Modbus was developed as a serial fieldbus by the company Modicon (today Schneider Electric) as a communication path between their control systems.

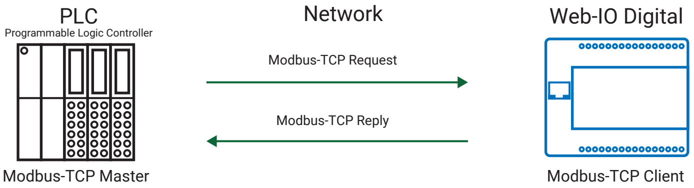
The clear and simple structure of the Modbus protocol has led other manufacturers to integrate Modbus into their devices. Modbus has thus developed into a standard that is still established today.

The master/slave principle

Modbus operates according to the master/slave principle. This means that there is at least one master and at least one slave.

Modbus slaves are e.g. PLC controls, Web IOs or other decentralized IO modules for digital and analog signals.

The master is always the communication partner who takes the initiative, i.e. sends a request, or the desired function call to a slave. Each slave has a unique address. The slave is normally purely passive and only responds if its address is specifically contacted.



With the increasing importance of TCP/IP Ethernet as a transmission option, the Modbus protocol was adapted almost 1:1 from serial data transmission to TCP.

Modbus-TCP works according to the client/server principle, whereby the master takes over the part of the client and the slaves act as servers. The Modbus master must therefore establish an explicit TCP connection to each Modbus slave. This connection is maintained for the duration of the communication. A new connection is not established for each request.

The standardized TCP server port for Modbus TCP is 502.

Information is exchanged by reading or writing memory addresses by the Modbus master, i.e. the client.

You can imagine this as if the Modbus slave, i.e. the server, had a cabinet with a lot of drawers, all of them numbered. Functions are assigned to the drawers.

If the Modbus master wants to retrieve certain information, it specifies the number of the corresponding drawer in its request and gets the content back from the Modbus slave.

If the Modbus master wants to trigger the slave to do something, e.g. operate a switching output, it places the necessary information in the drawer with the corresponding number.

As described earlier, this is actually done via corresponding memory addresses. A maximum of 65536 addresses are available. Which function is hidden behind which memory address is determined by the device manufacturer - i.e. it is not uniformly specified.

As a rule, the memory is divided into areas, separated according to function.

There are four different data types that can be accessed:

Description	**Data Typ**	**Access**	**Description**
Discrete Input	1-Bit	read only	digital input or switching status
Coil	1-Bit	read/ write	digital output or switching status
Input Register	16-Bit	read only	Value between 0 and 65535 or analog value or counter value
Holding Register	16-Bit	read/ write	Value between 0 and 65535 or analog value or counter value

Function codes are used within the Modbus protocol to specify which data type is to be accessed and how. Here is a list of the most common function codes:

FC (dec.)	FC (hex.)	Description
01	0x01	Read Coils Read digital outputs or switching outputs
02	0x02	Read Discrete Inputs Read digital inputs or switching states
03	0x03	Read Holding Registers Read 16-bit (output) registers
04	0x04	Read Input Registers Read 16-bit (input) registers
05	0x05	Write Single Coil Writing a single output or switching output.
06	0x06	Write Single Register Writing a single 16-bit register
15	0x0F	Write Multple Coils Writing several outputs or switching outputs
16	0x10	Write Multiple Registers Writing several 16-bit (output) registers
07	0x07	Read Exeption Status Request error status

Modbus-TCP Protocol structure

The Modbus TCP protocol frame has the following structure:

2 Bytes	2 Bytes	2 Bytes	1 Bytes	1 Bytes	n Bytes
Transaction ID	Protocol ID	Length	Unit ID	Function Code	Modbus Data

Transaction ID

The Transaction ID is something like a request number and is incremented by one by the master for each request. The client answers with the same Transaction ID.

Protocol ID

With Modbus-TCP always 0.

Length

Length of the Modbus data in bytes plus two.

Unit ID

In case of the serial Modbus protocol this was the address of the slave. The field was adopted for compatibility reasons. In case of Modbus-TCP, however, the unique addressing is done via the IP address of the slave.

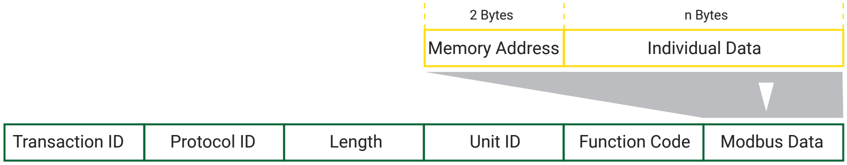
Function Code

The Modbus protocol defines via numbered functions codes, what the request sent by the master requires, i.e. which function is required of the slave.

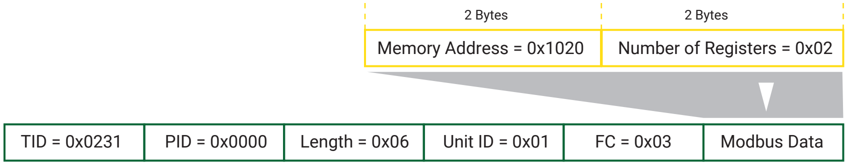
Modbus Data

The Modbus data area is filled with different contents depending on the used function code and can therefore be of different size. The data direction also plays a role in the structure of the Modbus Data area.

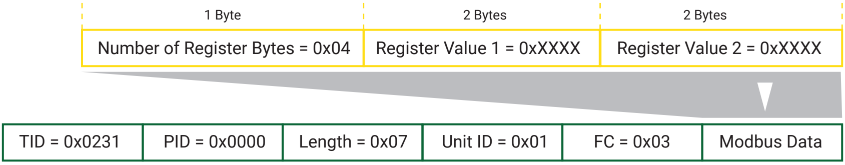
In case of data direction Master to Slave, the first two bytes always contain the memory address to be addressed.



The following example shows what a Modbus TCP packet looks like when retrieving two registers with Function Code 3 from memory address 0x1020.



The response packet is structured differently. Here the number of transferred register bytes is coded in the first byte of Modbus data. In the next 4 bytes are the contents of the requested registers.



Despite the straightforward protocol structure, Modbus-TCP offers great flexibility for industrial communication.

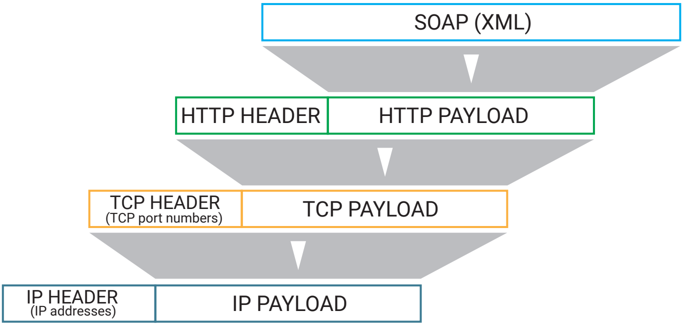
Tip: Pay attention to whether the device manufacturer specifies decimal or hexadecimal values when describing the memory addresses.

SOAP - Simple Object Access Protocol

SOAP is a web-based message protocol that is not only used in industrial environments.

Transmission at network level

SOAP is not bound to a special transport protocol. However, as a Web-based protocol, SOAP is generally based on HTTP or HTTPS. However, there are also rare applications where, for example, FTP or SMTP is used as the transport protocol. We will limit ourselves here to the description of SOAP in connection with HTTP(S).



The advantage of HTTP(S) as a basis for communication is that most networks, even if protected by firewalls etc., can be used for HTTP(S) throughout.

TCP ports 80 and 443 are used, as is usual with HTTP or HTTPS.

Due to the use of HTTP(S), SOAP works according to the client/server principle. The communication flow is always the same. The client sends an HTTP request to the server using the POST method. With the POST, the client transfers the corresponding data in XML format.

The server processes the transferred data and sends a corresponding confirmation.

The message format

As a message format SOAP uses the XML syntax. All information is embedded in XML tags. SOAP messages follow a predefined, structured format.

The actual SOAP part, which is enclosed by envelope tags, begins after the XML version is specified. The envelope tag contains as a parameter a reference to the fact that the standard SOAP format is being used and that the contents are coded accordingly.

Embedded in the envelope tags are a message header and the actual data.

The message header is optional and again enclosed by header tags. If it is used, it contains information on how to handle the actual data.

The data itself is enclosed by body tags.

```
<?xml version="1.0"?>
<soap:Envelope
 xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
 soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
  <soap:Header>
     ...
  </soap:Header>
  <soap:Body>
     ...
  </soap:Body>
</soap:Envelope>
```

The application specifies how the data within the body tag is structured and which tags are used.

Advantages of SOAP

- readable text
- supported by many software manufacturers
- independent of operating system and programming language

Disadvantages of SOAP

- High overhead, i.e. high data volume, since all transferred data is nested in XML tags.
- Binary content must first be converted into displayable text using MIME or Base64 encoding and then back again later.

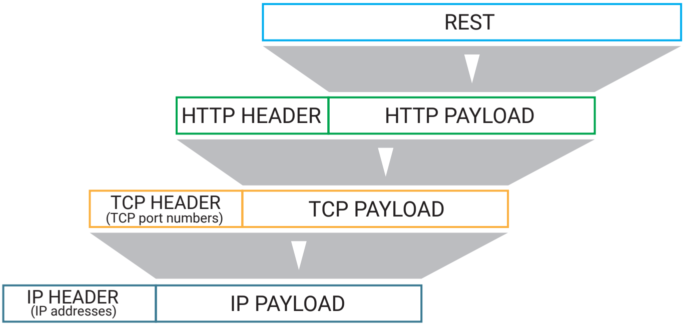
Further details on the SOAP standard can be found at: https://www.w3.org/TR/soap/

REST - REpresentational State Transfer

REST does not describe a concrete protocol structure. Instead, REST specifies properties that should be fulfilled for data exchange in an industrial environment.

Transmission at network level

It should be said in advance that REST uses HTTP or HTTPS as the higher-level protocol.



The advantage of HTTP(S) as the basis for communication is that most networks, even if protected by firewalls etc., can be used for HTTP(S) throughout. In addition, HTTP(S) meets the properties required for REST.

TCP ports 80 and 443 are used, as is usual for HTTP and HTTPS.

Properties and basic elements

Client/Server model

REST works according to the client/server procedure, whereby the action is always initiated by the client. The server provides resources (data, contents, functions). The client sends a request to access selected resources. With a reply, the server delivers the requested data or confirms the desired action.

Statelessness

With many client/server applications, a certain status (authorization, special task, specific purpose ...) is assigned on the server side when the connection is established and is retained for the duration of the connection. REST initially treats all transmissions to the server in the same way and only the content of the data transmission determines the further server-side handling or classification of the request. The status of an application is therefore in the hands of the client.

Stacked system

REST provides for a clear separation of responsibilities. This separation applies in particular to the handling of communication and the further processing of the transported contents.

This makes it possible to use proxies, gateways or similar intermediate stations on the transmission path.

In addition, HTTP or encrypted transmission via HTTPS can be selected for communication, depending on the security requirements. This makes no difference for the functional handling of the REST content.

Addressability of resources

All resources available on a server can be accessed via a unique address (URI = Uniform Resource Identifier). The URI is structured like the URL used for addressing in the browser:

```
Protocol://<Host>:<Port>/<Path>/<Resource>?<Parameter>&<Parameter>
```

Request methods

The standardized HTTP calls are available for the requests:

- GET retrieves resources from the server and is used read-only

- POST creates new resources or changes existing ones
- PUT changes existing resources
- DELETE deletes existing resources
- HEAD requests only the HTTP header, e.g. to check the availability of the resource to be retrieved
- OPTIONS retrieves information about the communication options

The retrieved data is called a representation of one or more resources. Hence the name REST for REpresentational State Transfer.

Ultimately, representations are nothing more than an image of parts of the process data. The transfer can take place in almost any form, but in a form to be agreed upon. Usually JSON, XML or raw text are used. But SVG, MP3 or other formats can also be used depending on the application. The representation can also include hyperlinks to further resources.

In the following example we see a GET request to query the sensor values to a Wiesemann & Theis Web Thermo-Hygrobarometer in JSON format:

The client sends:

```
http://10.40.22.19/rest/json/iostate/
```

And the Web Thermo-Hygrobarometer sends in response:

```
{
  "iostate":  {
    "sensor":[{
        "name":   "Temperature",
        "number":0,
        "unit":   "°C",
        "value":  25.9
      }, {
        "name":   "rel. humidity",
        "number":1,
        "unit":   "%",
        "value":  43.2
      }, {
        "name":   "Preassure",
        "number":2,
        "unit":   "hPa",
        "value":  994.7
      }]
  }
```

Code on Demand

Today it is common practice to load or reload e.g. JavaScript on websites. REST also allows you to reload a source code, program parts or function modules. This allows the client's function to be extended or changed during runtime.

Cache

REST is intended to answer repetitive requests from the client side from a cache in order to reduce the data load on the transmission paths. The server uses appropriate specifications in the HTTP header to determine whether or not the cache can be used for the requested representation.

Advantages of REST

Simple implementation, since the mechanisms of HTTP are used as far as possible.

Disadvantages of REST

Due to the request/reply technique, only polling, i.e. the targeted retrieval of data, but no event-controlled communication is possible. The relevant data must be continuously polled (queried) in order to detect changes.

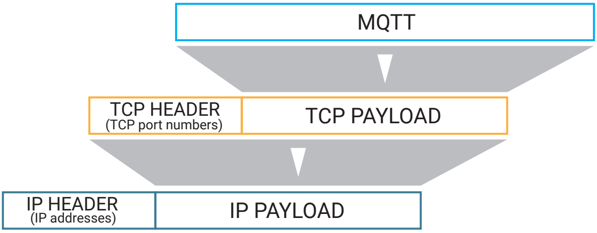
MQTT - Message Queue Telemetry Protocol

The special feature of MQTT is that two communication partners exchanging data with each other never do so via a direct connection.

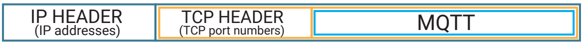
Instead, there is a central data intermediary, the broker. The broker receives data from one MQTT user and distributes it to others.

Transmission at network level

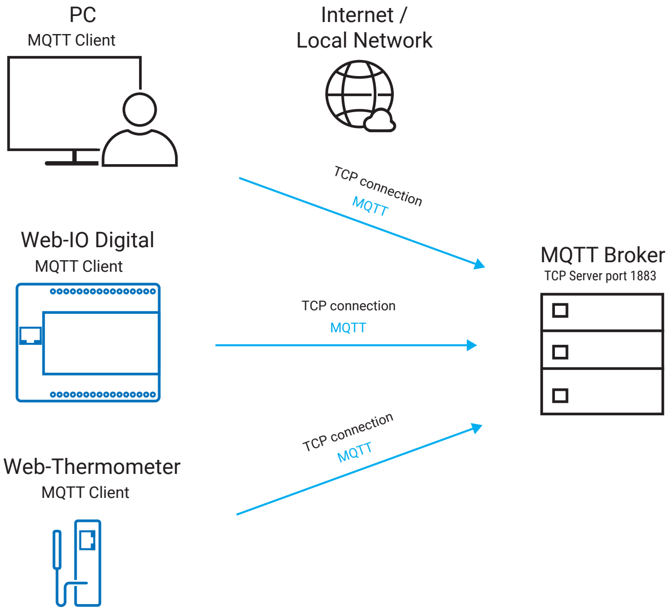
MQTT uses TCP as its basic protocol and thus works according to the client/server principle.



The MQTT data package therefore has the following structure:



The broker is the server that can accept connections on TCP port 1883 by default. The MQTT users act as TCP clients and connect to the broker as required.



Data exchange at protocol level

With MQTT there are two rules that a client can take on.

Publisher

As a publisher, the client sends data to the broker. This can be measured values, switching states or any other process data. In addition to readable text content, MQTT also allows binary data. Whether the publisher sends its data to the broker when a change occurs or cyclically depends on the application.

Subscriber

The subscriber accepts data from the broker. In the role of the subscriber, the client informs the broker after the connection has been established which data he wants to receive or subscribe to.

Each MQTT client can be a publisher, subscriber or both. So there is no mandatory master/slave principle as with other industry protocols. All MQTT end devices or clients initially have equal rights at the MQTT level. Who supplies and who receives data is therefore exclusively determined by the application through the publisher/subscriber assignment.

Topic

The MQTT broker manages the data to be exchanged according to data endpoints. The naming of the data endpoints is done via topics. These are strings, i.e. character strings, which can be structured in a similar way to the URL when calling up a website.

Example:

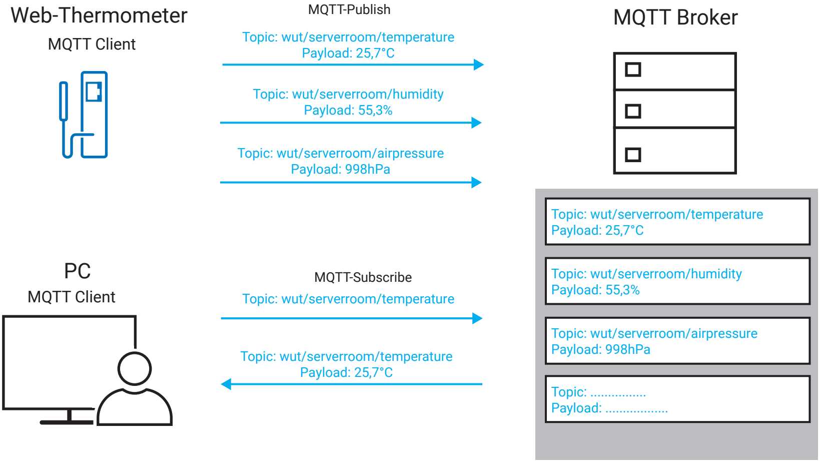
A W&T Web-Thermo-Hygrobarometer measures temperature, humidity and air pressure in the server room of the W&T company building. Each of the three values represents a data endpoint.

The topics could look like this:

```
wut/serverroom/temperature
wut/serverroom/humidity
wut/serverroom/airpressure
```

Via MQTT-Publish the Web-Thermo-Hygrobarometer transfers the measured values as payload, i.e. as data content, under these topics to the MQTT-Broker.

Any MQTT client can now send a subscription to the MQTT-Broker by specifying the desired topic. Here, the MQTT client sends a subscription to `wut/serverroom/temperature` and subscribes to the temperature value of the web thermal hygrobarometer.


Web-Thermometer
MQTT Client
MQTT-Publish
Topic: wut/serverroom/temperature
Payload: 25,7°C
Topic: wut/serverroom/humidity
Payload: 55,3%
Topic: wut/serverroom/airpressure
Payload: 998hPa
MQTT Broker
Topic: wut/serverroom/temperature
Payload: 25,7°C
Topic: wut/serverroom/humidity
Payload: 55,3%
Topic: wut/serverroom/airpressure
Payload: 998hPa
Topic: ....
Payload: ....
PC
MQTT Client
MQTT-Subscribe
Topic: wut/serverroom/temperature
Topic: wut/serverroom/temperature
Payload: 25,7°C

As long as the PC is connected to the broker as an MQTT client, it automatically receives the value sent by the Web-Thermo-Hygrobarometer via Publish.

The subscriber can work with wildcards “#” when specifying the topics.

Example `wut/serverroom/#` subscribes temperature, humidity and air pressure for the server room

In addition to “#” there’s a “+” wildcard. If “+” is specified, only end topics from the corresponding level are subscribed to - topics from underlying levels are ignored.

For each subscribed value there is a separate data transmission.

Data transmission is byte-oriented and thus binary transparent - any content can be transmitted. For standardization UTF8 is the default format for all text contents.

Special features

MQTT offers some special options for the transport and exchange of data, which can be set via flags for each connection or subscription.

Quality of Service – QoS

Values between 0 - 2 define the reliability with which a publish message is sent to the broker.

0 out and forgotten
The MQTT client does not expect any confirmation for sent data.
This procedure is insecure, but very fast.

1 at least once
The MQTT client sends the data several times if necessary until it receives an acknowledgment of receipt from the MQTT broker.
This procedure ensures that the data arrives at the broker, if necessary several times.

2 exactly once
Each data transmission must be explicitly acknowledged by the MQTT broker. This ensures that nothing is sent twice.
This procedure is the safest, but also the slowest.

In the case of QoS1 and 2, the question very quickly arises at first glance as to why a single data transmission is more secure than a multiple one. The following two examples help to explain why.

If a measured value is to be transmitted - e.g. a temperature - it does not matter whether the value arrives at a subscriber several times.

But if the angle by which a robot arm is to move is transmitted - e.g. 5° - and the data transmission arrives at the subscriber three times, the robot arm would move 15°, which could have fatal consequences.

Last Will and Testament

A publisher can specify that if the MQTT connection is lost, the broker sends a specific message to the subscriber(s) instead of the subscribed values/data.

Retained Message

By setting this flag, the publisher instructs the broker to cache the last value/data send and to transmit it immediately to a subscriber who reconnects.

All three features are especially useful when data is transmitted over transmission paths that are not always reliable (e.g. mobile networks).

Features and advantages of MQTT

MQTT is considered to be the transmission protocol for the "Internet of Things" and offers various advantages, especially for data exchange via the internet:

- Since all end devices using MQTT work as clients, overcoming firewalls and security measures is usually possible with little or no effort (no port releases and no NAT routing / port forwarding have to be set up in the firewalls).
- The publisher does not have to worry about which recipients actually receive the data provided.
- Unlike other internet-based protocols, binary data can be transmitted without Base64 or other encoding.

Peculiarities and disadvantages

- The data supplier does not know who actually receives his data (no end-to-end acknowledgment).
- No real-time capability

OPC – The process data translator

Basic information

In automation technology, hardware components from various manufacturers are usually combined to form a system. In the past, each manufacturer followed its own way of passing on process data to the software level. This applies both to the physical communication path and the data format.

To get around this process data chaos, the original OPC standard was introduced. The OPC Foundation, which was founded in 1996 as a non-commercial organization, was responsible for this. Members of the OPC Foundation are representatives of leading companies in the automation industry.

The aim was to create a globally accepted standard for communication in automation technology.

OPC DA - OPC Data Access

OPC DA is not a network protocol in the true sense. However, OPC DA is still an important industry standard and, depending on the end device, can also have points of contact with network communication.

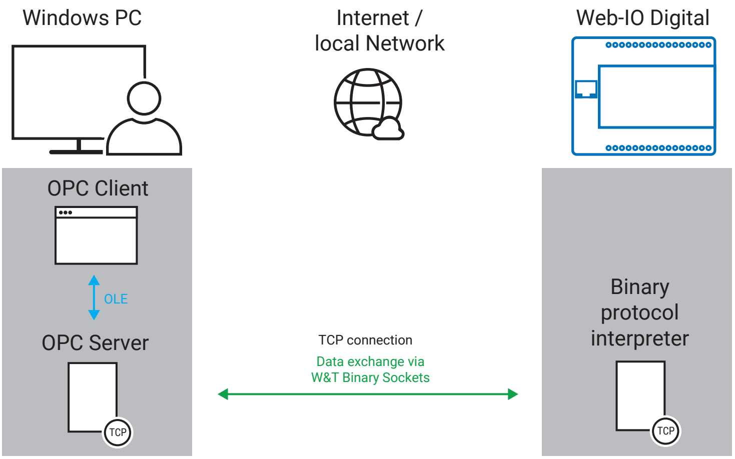
OPC stands for OLE for Process Control, where OLE is the abbreviation for Object Linking and Embedding. The basic idea of OLE is the controlled embedding of documents from other applications into your own application, for example, inserting an Excel document into a Word file.

Both OLE and OPC were designed specifically for PCs with Windows operating systems and only work on Microsoft operating systems.

OPC-supported applications do not communicate directly with the addressed end devices. Instead, an OPC server is installed for the corresponding end device. The OPC server is a software process that handles the manufacturer-specific communication with the terminal device in the background - similar to a driver for hardware on your own PC. The process data made accessible in this way is prepared according to the OPC standard and transferred to the application in a standardized form.

The part of the application that communicates with the OPC server is called the OPC client.

The following example shows the access to a Wiesemann & Theis Web-IO 12xDigital via OPC server:



Access via OPC

The original OPC interface is divided into four main tasks:

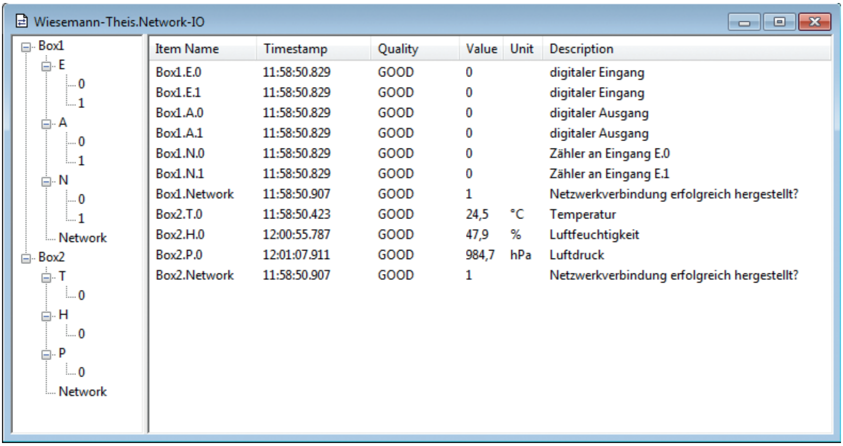
- Data Access: short DA, describes the exchange of real-time data via OPC.
- Alarm & Events: short AE, is used for alarm and event handling.
- Historical Data Access: HDA for short, allows stored, historical values and value histories to be made accessible.
- Data Exchange: DX for short, allows OPC servers to exchange data with each other.

OPC treats the process data as individual data endpoints. A data endpoint can be a measured value, a process status, a switching status and much more. The individual data endpoints are called items. The items can be read or written depending on their type.

All items have an item ID, an address that is unique within the OPC server. Each item has an undefined number of properties, or item properties, such as value, quality, time stamp, etc.

The items are usually grouped together by the OPC server. This results in a form of hierarchy (OPC-Server > OPC-Group > OPC Item).

To provide the OPC client with easy access to all available items, many OPC servers allow the OPC client OPC-browsing. This allows the OPC client to query all items in a type of directory tree structure. The following is an example of the item structures of a W&T Web-IO 2xDigital and a Web Thermo-Hygrobarometer.



Item Name	Timestamp	Quality	Value	Unit	Description
Box1.E.0	11:58:50.829	GOOD	0		digitaler Eingang
Box1.E.1	11:58:50.829	GOOD	0		digitaler Eingang
Box1.A.0	11:58:50.829	GOOD	0		digitaler Ausgang
Box1.A.1	11:58:50.829	GOOD	0		digitaler Ausgang
Box1.N.0	11:58:50.829	GOOD	0		Zähler an Eingang E.0
Box1.N.1	11:58:50.829	GOOD	0		Zähler an Eingang E.1
Box1.Network	11:58:50.907	GOOD	1		Netzwerkverbindung erfolgreich hergestellt?
Box2.T.0	11:58:50.423	GOOD	24,5	°C	Temperatur
Box2.H.0	12:00:55.787	GOOD	47,9	%	Luftfeuchtigkeit
Box2.P.0	12:01:07.911	GOOD	984,7	hPa	Luftdruck
Box2.Network	11:58:50.907	GOOD	1		Netzwerkverbindung erfolgreich hergestellt?

Communication between OPC client and OPC server

The OPC client can select a either a subset or all of the items offered by the OPC-Server and combine them into one or more groups. These groups need not be identical to the groups formed by the OPC-Server. The selected items are then subscribed by the OPC client in groups. This means that the OPC client does not have to constantly query the status of the items, but is automatically informed by the OPC server if one of the properties of an item changes. In this way the OPC-Server relieves the OPC client and thus the application.

When is it useful to work with OPC DA?

OPC is the ideal solution whenever a flexible application is to be created that needs to exchange data with the hardware of various manufacturers without great effort.

In applications of process control engineering and process and measurement data visualization, OPC is especially useful for the user.

Despite all the advantages of OPC technology, it is important to remember that programming a universal OPC client application is a complex task that requires a high degree of programming competence.

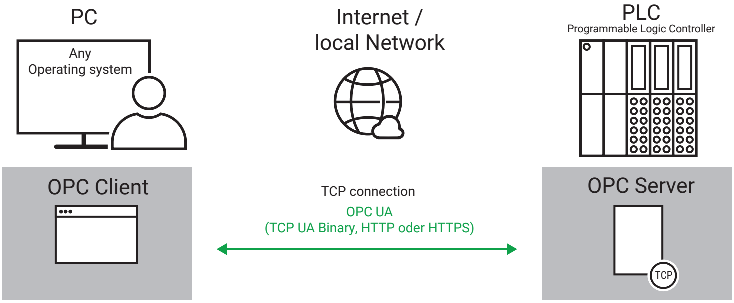
So, when it comes to creating a special application for a special terminal device of a manufacturer, one should consider whether it is not easier to take the direct communication path provided by the manufacturer.

OPC UA - OPC Unified Architecture

OPC UA is not an extended new edition of the original OPC standard. Instead, OPC UA follows a completely new concept and thus frees itself from many disadvantages that existed under the original OPC.

The concept of OPC UA

The most fundamental difference to the original OPC standard is that the OPC-Server no longer needs to be installed on the client side, so to speak as an additional driver. Instead, the associated OPC server works in every OPC UA-capable terminal device.



OPC UA is:

- platform independent:
  no longer bound to Microsoft operating systems
- scalable:
  System expansions are possible without installing additional OPC servers
- internet capable:
  due to TCP/IP as the basic protocol, OPC UA can be used across networks
- safe: If required, OPC UA can be secured by its own security mechanisms or SSL/TLS
  See *more about SSL/TLS in the chapter Data Security/Network Security.*

Data transfer at network level

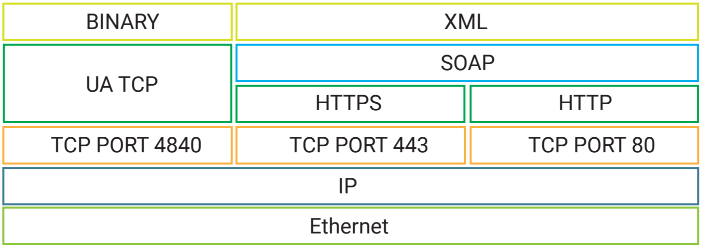
OPC UA works according to the client/server principle. To outsource the OPC server to the terminal device, standardized communication is required on the transmission path between OPC client and server.

To ensure this, TCP/IP was chosen as the basic protocol and Ethernet as the physical standard.

OPC UA distinguishes between three transmission variants:

- **HTTP**
  Data is sent or requested via HTTP requests.
  Information is transferred in SOAP or XML format.
  TCP server port is 80.
- **HTTPS**
  For HTTPS the same applies as for HTTP, but HTTPS work with SSL/TLS encryption.
  TCP server port is 443.
- **UA TCP Binary**
  The binary variant dispenses with the overhead caused by the additional XML tags. Instead, there is a very lean protocol that regulates the data exchange. This makes the data exchange much faster.
  TCP server port is 4840.

OPC UA PROTOCOL LEVEL



OPC UA already offers very flexible access options at network level.

Protocol and application level
Another new feature is that OPC UA allows access to individual items as well as to complex data structures. In addition, OPC UA can be used to call programs and functions on the end device itself.

The original standards OPC DA, EA, HDA and DX have been integrated into OPC UA as possible application options.

The OPC UA Server

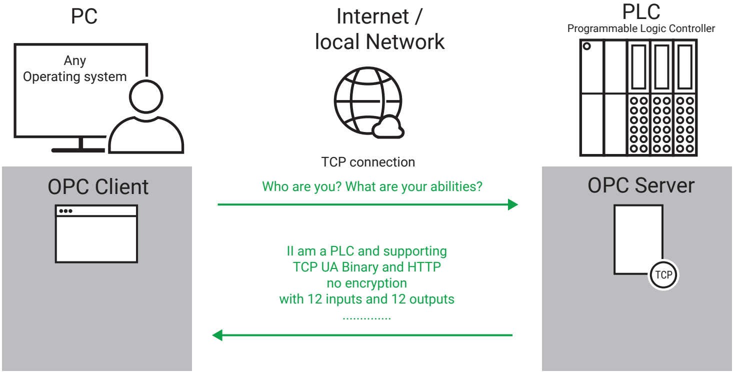
The main feature of OPC UA is, as already explained, that the OPC server is part of the terminal device to be addressed. Even though OPC UA offers a wide variety of possibilities, not every OPC server has to support the full range of possibilities. It is sufficient if the server can handle the subset required for the application.

In the form of standardized type information, the OPC server summarizes which options and protocol variants it supports.

The OPC UA Client

In contrast to the OPC UA server, the OPC UA client should support as many of the different variants as possible. This is the only way to ensure that a high degree of compatibility with as many end devices as possible be achieved.

The OPC UA client can retrieve the type system information from the OPC UA server. This information records which transfer methods, items, variables, objects, functions, etc. are available. This considerably simplifies the integration of new end devices and the associated configuration effort.



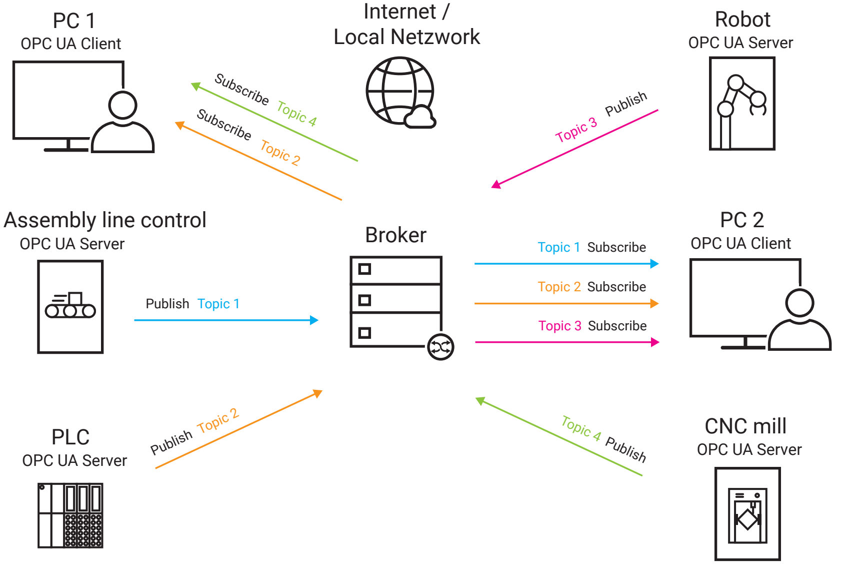
In addition to OPC UA, many clients also support the original OPC standard, so that mixed operation is also possible. For end devices that do not support OPC UA there are some providers of OPC UA gateways that will integrate these end devices by default into OPC UA applications..

OPC UA Pub/Sub

In 2018 the OPC Foundation published a new release of the OPC UA standard. OPC UA-Pub/Sub maintains full compatibility with OPC UA, but supports the Publish/Subscriber-Model. For this purpose OPC-UA-Pub/Sub uses the mechanisms of MQTT.

In end devices that use the Publish/Subscriber procedure of OPC UA, the OPC server is extended by an MQTT client service. Even if it is linguistically somewhat misleading, the term OPC server is still used.

OPC client and OPC server can send data to a broker via Publish as well as subscribe to data via Subscribe.



This allows process data to be passed on to a large number of end points with little effort.

Another new feature of the OPC UA Pub/Sub release is that UDP is also permitted as a basic protocol. Since UDP is faster than TCP due to a lower overhead and connectionless communication via datagrams, the use of UDP is particularly advantageous for applications that depend on short response times.

Possible use cases

Where the original OPC applications were concerned, it was usually the case that an OPC client, as the central control system, monitored and, if necessary, also controlled the participating end devices. The control system almost always provides an access option for the user via screen and keyboard.

In addition to this classic variant, OPC UA also supports the following communication models (without the involvement of a control system):

- Device to Device
- Device to Data base
- Device to Cloud

This makes OPC UA much more flexible than the classic OPC.

Data security / Network security

Basics

Safety requirement

How much security is required for data transmission? There is no general answer to this question. The demand for data security varies from case to case and can be very individual depending on the application and network environment.

But what criteria must be met for data transmission to be classified as secure?

Specifically, there are three requirements that must be met:

- Integrity
  If the data is changed or manipulated during transport, this must be detected immediately.
- Confidentiality
  Third parties must not be able to read the transmitted data contents.
- Authenticity
  It must be ensured that the communication partner is actually the one with whom you want to exchange data.

Other important aspects are:

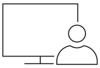
- Availability
  Both the services offered by the communication partners and the infrastructure required for communication should be usable at all times.
- Controllability
  Even if the technology required for data security is very complex, it should remain manageable for users and administrators alike.

In the following sections we will explain step by step the basic techniques that are used in practice to fulfill these points. Finally, we will summarize the combination of these techniques using HTTPS (secure communication in the browser).

Terms and symbols

We will confine ourselves to describing the principles of security measures without explaining the very complicated mathematics behind them.

We will work with the following symbols:



Client

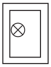
The client is the one who takes the first initiative in data communication. When surfing the internet, for example, the browser.

Server

A server offers data services that can be used by one or more clients. For example, a web server from which web pages can be called up and with which the client can exchange data.

Communication Data

Data exchanged between client and server



Safe

The safe symbolizes data secured by encryption.

Symmetrical key to encrypt and decrypt

In data technology, the key stands for a numerical value that is used for encryption and decryption. In symmetrical encryption, the same key is used for encryption and decryption.

Private Key for encryption

Public Key for decryption

With asymmetric encryption, different keys are used for encryption and decryption. The key for encryption (locking) is symbolized as a padlock, which can only be opened with the corresponding key.

Public Key for encryption

Private Key for decryption

In addition, one of the two keys is always public, i.e. accessible to everyone - only the owner knows the other private key. Whether the key for encrypting or the key for decrypting is the public key depends on the application (more on this in the following).

Key pairs that belong together are shown in the same color.

Hash value calculation

A kind of checksum calculation over a data set

Hash value

Symbol for the hash value

Certificate Authority

Certificate Authorities (CAs) issue certificates and confirm the authenticity with a kind of digital signature.

Certificate

Certificate data

Content of the certificate

Digital Signature

The signature is a type of digital signature that ensures the authenticity of a certificate.

Signed certificate

Signed certificate including the public key of the certificate owner.

Communication data

What is data actually?

In order to understand how data security works, we must first of all remember the form in which our data is encoded and transmitted.

No matter whether text, web pages, pictures, music, videos or other data are to be transferred - a certain amount of bytes is always transferred from A to B.

Bits and Bytes

As a reminder: At the lowest level, computers work with bits, i.e. with memory locations that can have the value 1 or 0. Eight bits make one byte.

A byte is a numerical value between 0 and 255. In data technology, bytes are usually represented in two-digit hexadecimal notation - i.e. 00 to FF (see chapter Number Systems).

Coding

Depending on the application, for example, a text becomes a certain amount of bytes, with each byte corresponding to one letter.

S	E	C	U	R	I	T	Y	*Text*
53	45	43	55	52	49	54	59	*Bytes ASCII-kodiert*
Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	*(hexadezimal)*

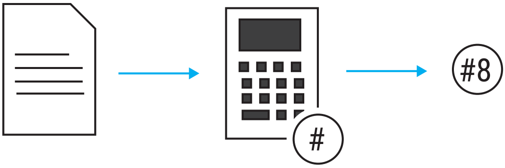
In a picture, a set of bytes would encode which particular pixel has a particular color in a particular position.

The significance of the individual bytes in the application is not important on the transport route. Here it is only a corresponding amount of bytes, i.e. numbers with which one can perform arithmetic operations if required.

Integrity - checksum and hash value

The principle of hash values

To ensure that transmitted data has not been altered in transit, the sender provides it with a kind of fingerprint. In practice, this is done by mathematically calculating the data in its entirety or in blocks. The result is called the checksum or hash value.



The trick here is that the original data cannot be calculated back from the checksum. Such calculations are called one-way functions.

A very simple example of one-way functions is the modulo calculation, i.e. the calculation of the residual value in a division.

Example:

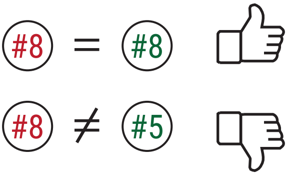
I divide the original value, e.g. 36, by 7. The result is 5 and there is a remainder of 1. The 1 would be the hash value in this case.

If you only have the residual value, it is impossible to clearly name the original value from which this residual value has resulted. This simple procedure is of course unsuitable for ensuring the integrity of data, since there are a large number of original values that produce an identical hash value. Therefore, security protocols use much more complicated arithmetic operations.

Hash values in practice

To ensure that the data has not been altered in transit, the sender calculates the hash value of his data transmission and informs the recipient. The recipient in turn calculates the hash value of the received data.

Only if both hash values are equal can it be assumed that the data has arrived unchanged.



In the following, the most common standards for hash value calculation are briefly introduced. If you do not want to delve deeper into the standards, you can read on in the next section.

- **MD5 - Message Digest Algorithm 5**
  A common method for checksum calculation is the MD5 algorithm. The MD5-Algorithm generates from any number of bytes a result that is always 128 bit long, the MD5-Hash. If even a single byte is changed in the existing data volume, this leads to a completely different MD5 hash.

  However it was found that by putting in considerable computational effort it was possible to find, or generate different original data, which resulted in the same MD5 hash. Once these so-called collisions became known, MD5 was no longer considered to be completely safe.

- **SHA-1 - Secure Hash Algorithm 1**
  After doubts about the security of MD5 arose, some competing methods for hash value generation emerged, but none of them have caught on. With the increasingly global use of the internet, SHA-1 was then defined as a uniform standard in the early 2000s.

  SHA-1 provides a 160 bit long hash value and is still considered safe today, even though it could be theoretically proven that collisions could also occur with SHA-1 if enough computing power was available.

- **SHA-2, SHA-3 respectively SHA256**

In order to further increase the level of safety, the SHA-2 and later even SHA-3 procedures were introduced. Both standards allow different bit depths for the hash value (SHA224, SHA256, SHA384 and SHA512). SHA256, which works with 256-bit hash values, is the most common.

It is important that sender and receiver agree on the same hash procedure.

Confidentiality through encryption

Confidentiality is one of the pillars of secure data transmission. For this purpose, the transmitted data must be protected from access by third parties during transport.

Encryption is an effective protection against data spying.

Encryption in data technology means a mathematical/logical manipulation of the transmitted bytes or numerical values.

The transmitted information is thus only readable for the intended recipient, who knows both the encryption method and the key used. All others see only incomprehensible number confusion.

There are two ways to achieve this:

1. **Security by Obscurity**
   Security through obscurity - The data is encrypted using a secret procedure or algorithm. Only those communication partners who know the procedure used can then exchange data.

   Depending on the algorithm used, this could provide the desired security. Since open source software is increasingly being used, especially for internet applications, the algorithms used would also be very easy to spy out. Security by Obscurity is therefore not suitable for the requirements of today's internet applications.

2. **Full Disclosure**
   Full disclosure - In this case, the data is encrypted according to standardized procedures. The security of the data is achieved by the applied algorithm including secret numerical values - so-called keys - in addition to the transmission data. The communication partners can thus agree on a generally known procedure and use (numerical) keys that are not accessible to unauthorized parties.

   Using this method, even today's internet applications can exchange data with any number of communication partners.

The common encryption methods and the key exchange among each other are presented in the following sections.

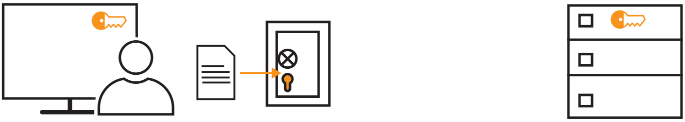
Symmetric encryption

As already explained, encryption in data technology works by mathematical/logical manipulation of the transmitted bytes or numerical values.

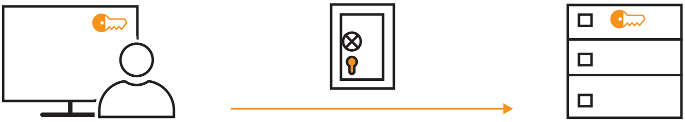
In symmetrical encryption, the sender and receiver use an identical key that is used for both encryption and decryption.

The principle of symmetrical encryption

The following analogy may help. Simply imagine that important documents are being packet in a lockable safe for transport.



Before shipping, the safe would be locked and sealed by the shipper with the appropriate key and sent off on the transport route.



The recipient has the same key and can unlock it to access the content.

How Symmetric Encryption Works

When data is transmitted, it is of course not enclosed in a container.

A very simple encryption could look mathematically like adding a number x to each byte. The number x would be the key in this example.

The recipient could in turn subtract the key value x again, i.e. apply the arithmetic operation in reverse to restore the original data.

Of course, such a simple algorithm would be much too easy to crack and is therefore unusable in practice. But the example shows the basic principle of encryption.

Arithmetic operations used for real encryption are much more complex and do not just manipulate single bytes.

Symmetric encryption is the term used to describe procedures in which the sender and recipient of a message use the same secret key when encrypting and decrypting, as described in the previous example.

A basic distinction is made between block encryption and stream encryption.

Block encryption

In the case of block encryption, the data to be transmitted is divided into data blocks of equal size. If not enough bytes remain for the last data block, additional fill bytes are added.

Each data block is encrypted separately. The bytes of a block are not only replaced (substitution) but their position within the block is also swapped about (permutation). Depending on the method used, already encrypted data blocks are re-encrypted several times in succession.

Furthermore, block encryption distinguishes between three encryption modes:

- **ECB-Modus - Electronic Code Book Mode**
  All blocks are coded in the same way only by means of the key. The encryption of blocks with the same content always results in the same encrypted code.

- **CBC-Modus - Cipher Block Chaining Mode**
  In addition to the key, parts of previous blocks also flow into the coding. Encoding two blocks with the same content does not lead to the same result if the

preceding blocks are different.

- **GCM - Galois Counter Mode**
  In addition to the key, a continuously running counter is included in the encryption. Here, too, two blocks of the same content produce completely different data after encryption. One advantage of this technique is that the blocks do not have to be encrypted one after the other, as in CBC. Instead, several data blocks can be encrypted simultaneously. This makes GCM much faster than CBC.

Symmetrical block encryption almost exclusively uses the GCM mode.

Block encryption is used, for example, for secure e-mail transmission and the transmission of web pages via HTTPS. However files stored on the hard disk can also be block encrypted to protect them from unauthorized access.

Stream encryption

In stream encryption, each byte transmitted is encrypted individually. A key stream is generated parallel to the user data stream.

With the help of the common key, a sequence of randomly appearing bytes is formed using a predefined algorithm. Byte for byte, a logical-mathematical link is then created between the user data stream and the key stream. The ensuing result is then transmitted and decrypted by the receiver in reverse.

In addition to the key stream, Some procedures also include in the calculation bytes of the user data stream that have already been transmitted.

Stream encryption is mainly used for the transmission of analog data and video signals, because block-by-block processing could lead to blockages in the continuous data flow.

Symmetric encryption standards

In addition to a common key, the two communication partners must agree at the beginning of the data exchange on the standard to be used for encryption.

Here are the most common standards:

- **AES - Advanced Encryption Standard**

The encryption algorithm AES is the most widely used method for symmetrical block encryption. Key lengths of 128, 192 or 256 bits can be used optionally. The possible block sizes are between 128 and 256 bits. Each block is encrypted up to 14 times in a row.

The Rijndael algorithm used (named after its inventor) is an open standard, works fast and efficiently and is therefore also well suited for less powerful hardware.

- **DES - Data-Encryption-Stadard**
  DES was developed by IBM as early as the 1970s and is still used today, although AES has in the meantime become established as its successor.

  DES works with 56bit keys and a block size of 64 bit.

  Because its 56 bit keys are rather short , DES is no longer considered secure today, because with today's computers it is possible to find the right key by brute-force attacks (trying out all possibilities).

- **RC4 - Rivest Cipher 4**
  Due to its manageable code base, simple integration and low resource requirements, RC4 has long been one of the most popular encryption methods.

  Today the RC4 power encryption is considered insecure.

- **ChaCha20**
  ChaCha20 is a power encryption system developed by Daniel J. Bernstein. It works with a key length of 256 bits. An advantage of ChaCha20 is the very high encryption speed.

- **Twofish**
  Twofish works with blocks of 128bit and key lengths of 128 bit, 192 bit or 256 bit. Although Twofish is considered secure, it is hardly used in data transmission.

There are a few more encryption methods, but they are of little practical relevance today.

Preshared Keys

A disadvantage of symmetrical encryption is that both communication partners need the same common key. The keys must therefore be made available to both communication partners before data is transmitted. The problem with this is: How can you ensure that both sides receive the same key, but that this key remains secret at the same time?

Despite the key problem - symmetric encryption methods are very fast and therefore particularly suitable for the transmission of large amounts of data.

Asymmetrical encryption

Private and public keys

As a reminder, the biggest security problem with symmetric encryption is that the common key must be transmitted to both communication partners while at the same time being protected from spying.

Asymmetric encryption therefore works with two unequal keys, which, however, belong inseparably together and can only be used together as a key pair.

One of the keys is public. It can be seen by everyone and therefore does not need to be protected against spying on its way to the user. This key is called the public key.

Public Key
for encryption

The other key is top secret and known only to its owner.

Never let this private key out of your hands.

Private Key
for decryption

Both keys of the key pair are generated by the user of the private key. This is the only way to ensure that the private key is kept secret, since it never has to leave the owner's area of responsibility.

Only the public key is passed on.

Procedure of asymmetrical encryption

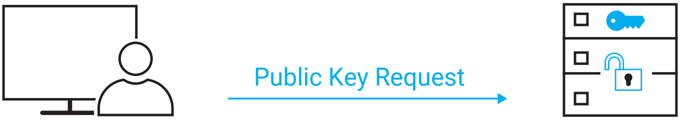
In the case of asymmetrically encrypted data transmission, the public key is used for encryption. You can think of it as a padlock which is sent open and which the sender simply snaps shut to encrypt it.

The data encrypted in this way can only be decrypted with the corresponding private key. The private key cannot be used for encryption in this case.

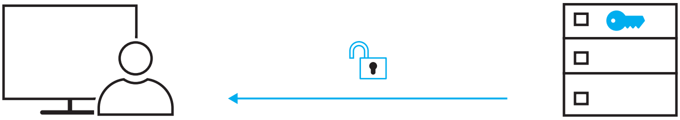
There are use cases where it works exactly the other way around - more on this later.

In the following, the principle procedure of asymmetrically encrypted data transmission is shown step by step:

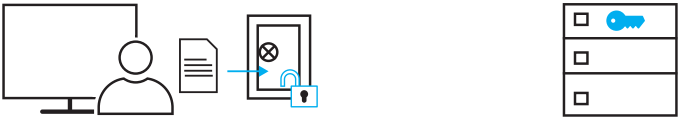
The sender requests the public key - i.e. the opened padlock - from the recipient.



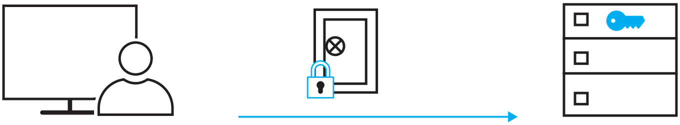
He can transfer the public key to the sender without any risk, as it can only be used to lock the safe, i.e. to secure the data.



The sender encrypts his data with the public key - puts it in the safe, so to speak - and snaps the lock shut.



Once the lock is attached to the safe and snapped shut, not even the sender can access the secured data.



Only the recipient who is in possession of the top secret private key can decrypt the data again - i.e. reopen the safe using his private key.

Only the legitimate recipient has access to the original data.

Functionality of asymmetric encryption

The mathematical procedure behind the real data technology is much more complicated.

Here again, the keys are numbers that are mathematically interdependent, but nevertheless the private key cannot be calculated with knowledge of the public key.

The complex algorithms and the size of the numbers chosen as keys mean that encryption and decryption require a great deal of computational effort and are therefore very time-consuming. A purely asymmetric encryption is therefore only suitable for smaller amounts of data. Even current computer hardware would reach the limit of available computing power for larger data streams.

Standards for asymmetric encryption

There are currently three common asymmetrical encryption methods:

- **RSA - Rivest, Shamir und Adleman**
  The cryptographers Ron Rivest, Adi Shamir and Leonard Adleman developed the RSA procedure as early as 1977. The key length is variable with RSA. For a secure connection, the Federal Office for Information Security recommends keys of at least 2048 bits.

- **Diffie-Hellman**
  The Diffie-Hellman method is not an encryption method in the true sense. In 1976, the cryptographers Whitfield Diffie and Martin Edward Hellman developed an algorithm to agree on a common key between two communication partners, which can then be used for symmetrical encryption. Even though this procedure is called key exchange, the actual key is not transmitted. Instead, the algorithm

used allows both sides to calculate the common key.

- **ElGamal**
  The ElGamal encryption method was developed in 1985 by cryptologist Taher ElGamal and is based on the methods already used by Diffie-Hellman. However, the algorithm was changed in such a way that, in addition to the pure key exchange, user data can also be encrypted and sent. ElGamal is not subject to patents and is therefore often used in open source projects.

Elliptic Curves
Cryptography based on elliptical curves is not a further proprietary encryption method. In simple terms, all asymmetric procedures work with one-way functions that process very large prime numbers. This requires a lot of computational efforts and is time-consuming. In layman's terms, points on an elliptical curve are used instead of prime numbers. This method is much faster and can be applied in principle to all asymmetric encryptions mentioned above.

Hybrid Encryption

In order to be able to send even large amounts of securely encrypted data, in practice both methods are combined. This means that in the first step the asymmetric encryption method is used to exchange the secret key for a symmetric encryption. In the second step, the user data to be transmitted is encrypted with the common symmetric key.

Here is the complete procedure again using the example of a confidential connection between client and server:

The server has a pair of keys for asymmetrical encryption.

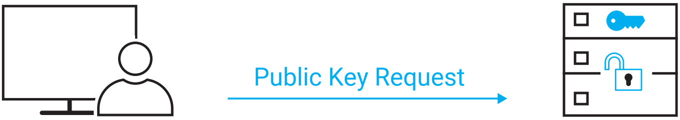
🔓 Public Key
for encryption

🔑 Private Key
for decryption

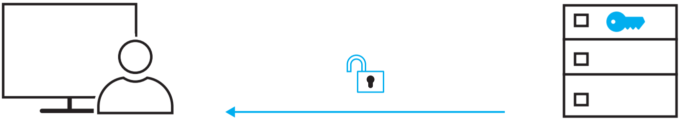
The client generates a symmetrical key when the connection is established.

🔑 shared
symmetric key
to encrypt and decrypt

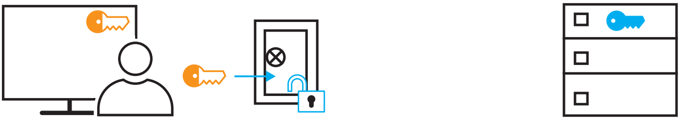
In the first step the client establishes an unencrypted connection to the server and requests the public key from the server.



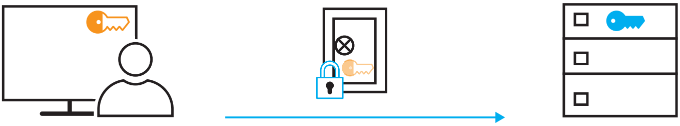
The server then sends its public key to the client.



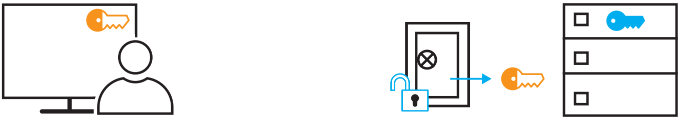
The client encrypts the symmetric key it has using the public key of the server.



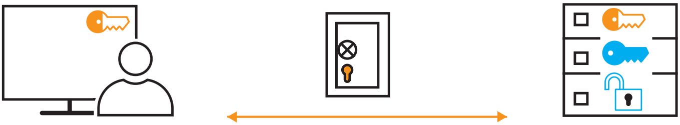
The key is protected from being read by third parties and is securely transmitted to the server.



The server can decrypt the symmetric key using its private key.



Now client and server have the common key for symmetrical encryption and can exchange data quickly and securely.



While the asymmetric key pair of the server normally always remains the same, the symmetric key of the client is only valid for one connection cycle. For a later data exchange, the client generates a new key.

Key calculation according to Diffie-Hellman

Encryption without key transfer

As we have already stated, the most critical point in symmetrically encrypted data transmission is that the communication partners receive the same secret key without it being spied out in transit.

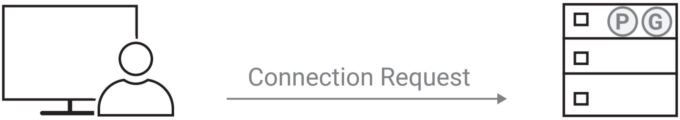
In case of the Diffie-Hellman method, neither the common key nor parts of it are actually transmitted. Instead, in layman's terms, numerical values are exchanged from which each communication partner on its side can calculate the common key from there side.

How Diffie-Hellman works

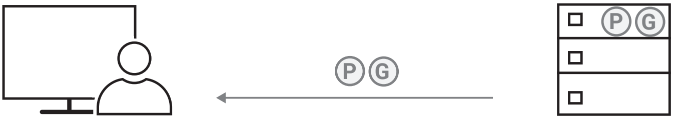
The Diffie-Hellman method requires a very large prime number P and a positive integer G, which must be smaller than P.

Ⓟ	very large prime number
Ⓖ	smaller integer

Both communication partners need this pair of numbers. It is generally transferred from the server to the client, which requests it when the connection is started.



P and G are not secret and can be passed on through unsecured channels without hesitation.



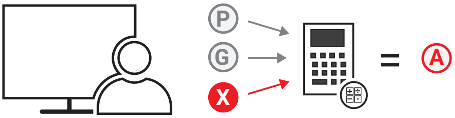
In addition, each of the two communication partners generates another integer that only they themselves know (X and Y).

Y secret integer generated by the server

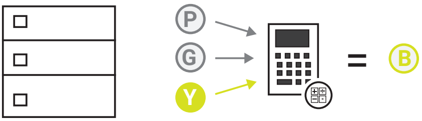
X secret integer generated by the client

So both communication partners have three numbers each: two public and one secret.

Now the client calculates another number A from the public numbers P and G and the clients secret number X according to a predefined one-way algorithm.

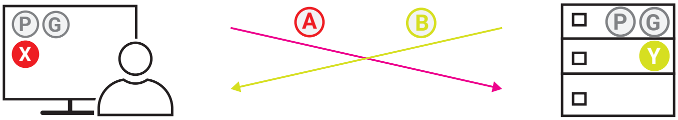


The server uses the same algorithm to calculate a number B.



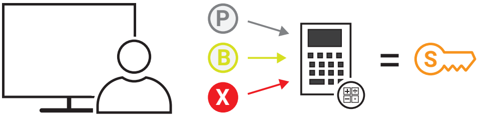
Since it is a one-way algorithm, the secret numbers X and Y cannot be unambiguously calculated back from the public numbers P and G and the result A and B respectively.

Client and server can therefore exchange A and B with each other without protection.

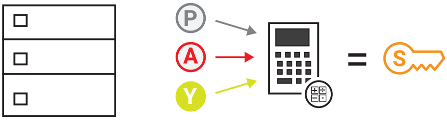


This means that client and server each have four numbers of which three are needed for further processing.

The client calculates the common key S from the public numbers P and B and its secret number X according to a further specified algorithm.



The server uses the same algorithm for P and A and its secret number Y.



The algorithms used are structured in such a way that both sides end up with the same result and thus with the same common secret key.

The key calculated in this way is used in the further course of the connection for symmetrically encrypted data transmission. All numbers used up to that point are no longer needed for actual encryption.

The mathematics behind Diffie-Hellman

In this section, we will use a simple example to show the algorithms used in Diffie-Hellman key calculation. We have deliberately omitted a detailed explanation of the formulas shown.

The Diffie-Hellman key exchange uses extremely large prime numbers. This is the only way to ensure that the calculated key cannot be found out by trying out different numerical values.

In our example we are working with the smallest possible values, which of course would not be safe in practice.

P	= 5	prime number
G	= 4	smaller integer
X	= 3	secret integer of the client
Y	= 2	secret integer of the server

First, client and server calculate the numbers A and B from the public numbers P and G and the secret numbers X and Y respectively.

The formula for the client:

$$A = G^{X} \bmod P$$

$$A = 4^{3} \bmod 5 = 4$$

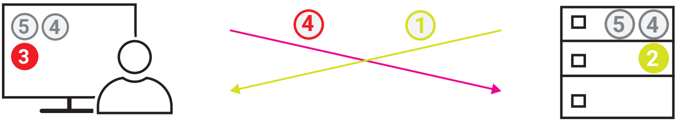
Remember: mod stands for modulo and is the remaining calculation for the division

The formula for the server:

$$\textcircled{B} = \textcircled{G}^{\textcircled{Y}} \bmod \textcircled{P}$$

$$\textcircled{B} = 4^{2} \bmod 5 = 1$$

The two calculated numbers are exchanged.



Now both sides independently calculate the common and secret key.

The formula for the client:

$$\textcircled{S} = \textcircled{B}^{\textcircled{X}} \bmod \textcircled{P}$$

$$\textcircled{S} = 1^{3} \bmod 5 = 1$$

The formula for the server:

$$\textcircled{S} = \textcircled{A}^{\textcircled{Y}} \bmod \textcircled{P}$$

$$\textcircled{S} = 4^{2} \bmod 5 = 1$$

Both calculations lead to the same result. In this example, the common key for the symmetric encryption now beginning would be 1.

Diffie-Hellman in summary

The Diffie-Hellman method is not an encryption method in the true sense. In 1976, the cryptographers Whitfield Diffie and Martin Edward Hellman developed an algorithm to agree on a common key between two communication partners, which can then be used for symmetrical encryption. Even though this procedure is called key exchange, the actual key is not transmitted. Instead, the algorithm used allows both sides to calculate the common key.

Diffie-Hellman Elliptic Curves

Diffie-Hellman key calculation on the basis of elliptical curves is not a separate procedure. In layman's terms, points on an elliptical curve are used instead of prime numbers. This method is not as computationally intensive and therefore somewhat faster than the classic Diffie-Hellman key calculation.

Authentication through certificates

First the definition of three terms that are often confused:

Authentication
means proving one's identity.

Authentification
means checking someone else's identity for validity.

Authorization
is the transfer of rights reserved within a system to certain persons or institutions.

In the previous sections we have learned about ways to protect data from being changed or read during transport. But what use is this security if the communication partner with whom I exchange data is not the one I actually want to communicate with? Important data may end up at the wrong recipient.

This security problem can only be solved if the communication partner can be clearly identified.

In real life, when I want to know who I am dealing with, I ask my counterpart for their identity card. This is issued by a trustworthy authority and if the picture, name and address identify the right person, everything is fine.

Certificates

In data transmission there are also such identity cards, but they are called certificates.

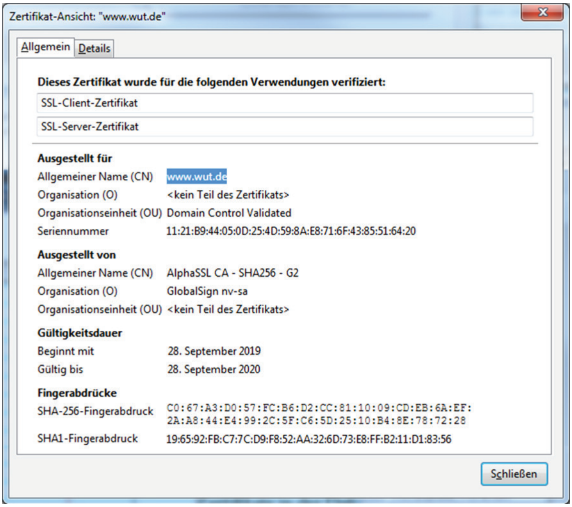
Certificate holder
Who needs a certificate?

All institutions and persons who want to operate secure data services such as web servers need a corresponding certificate to identify themselves to the user.

What information does a certificate contain?

- Node (for whom is the certificate issued?)
- Name or IP address of the server for which the certificate is valid
- Public key of the node
  (to switch to secure encrypted communication later)
- Intended use
  (for which data service was the certificate issued?)
  - Web server authentication
  - Software signature
  - Authorization to issue interim certificates
  - Authentication of mail servers
- Depending on the application
  - IP address
  - Domain or host name
- Serial number
- Expiry date (how long is it valid?)
- Certificate issuer (who issued it?)
- Signature of the certifier

Here is as an example, the certificate information for the Wiesemann & Theis web server:



Dieses Zertifikat wurde für die folgenden Verwendungen verifiziert:

SSL-Client-Zertifikat

SSL-Server-Zertifikat

Ausgestellt für

Allgemeiner Name (CN)	www.wut.de
Organisation (O)	<kein Teil des Zertifikats>
Organisationseinheit (OU)	Domain Control Validated
Seriennummer	11:21:B9:44:05:0D:25:4D:59:8A:E8:71:6F:43:85:51:64:20

Ausgestellt von

Allgemeiner Name (CN)	AlphaSSL CA - SHA256 - G2
Organisation (O)	GlobalSign nv-sa
Organisationseinheit (OU)	<kein Teil des Zertifikats>

Gültigkeitsdauer

Beginnt mit	28. September 2019
Gültig bis	28. September 2020

Fingerabdrücke

SHA-256-Fingerabdruck	C0:67:A3:D0:57:FC:B6:D2:CC:81:10:09:CD:EB:6A:EF: 2A:A8:44:E4:99:2C:5F:C6:5D:25:10:B4:8E:78:72:28
SHA1-Fingerabdruck	19:65:92:FB:C7:7C:D9:F8:52:AA:32:6D:73:E8:FF:B2:11:D1:83:56

Certificate authorities

Certificates in data transmission are of course not issued by the local authorities like a form of identification would be.

Instead, there are trustworthy certification authorities - Certificate Authorities or CAs for short.

Certificate Authority

The issuing, the structure of certificates and how they are further used is laid down in the ITU-T standard X.509. This is why they are often referred to as X.509 certificates.

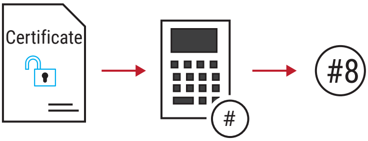
X.509 certificates are only valid if they are signed by the issuer - i.e. they are provided with a digital signature, or a digital fingerprint.

Signing Certificates

The CAs sign certificates in two steps

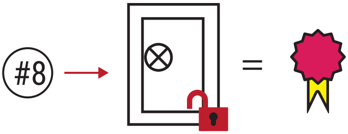
Step 1

A hash value is formed over the entire certificate content using a selected mathematical procedure.



Step 2

The CA encrypts the hash value and attaches it to the certificate.



The hash value is encrypted asymmetrically. However, in this case, the private key of the CA is used for encryption and the public key is used for decryption.

Private key of the CA

Remember: With normal asymmetric encryption it was the other way round - public key encrypted / private key decrypted.

The node receives the signed certificate, which consists of the unchanged certificate data and the signature, i.e. the encrypted hash value.

Distribution and validity of certificates

A decisive prerequisite for the security provided by X.509 certificates is that the private key of a CA remains secret and never gets out. For this reason, the CAs are not connected to the internet in order to protect the private keys from spying.

In contrast, the public keys of the root CAs (public keys) must be made available to all users as easily as possible.

Publishers of software and operation systems have archived this by programming in frequently required keys in advance or store them in a trust store (secure storage).

In addition, applications and operating systems usually offer the possibility of reloading certificates and recognizing them as trustworthy.

The import and recognition of certificates should be carried out with great care.

However, the identity of a server or an application cannot be guaranteed solely by the correct signing of the associated certificate. In the next step, the other data of the certificate must be checked.

The procedure for using X.509 certificates is roughly as follows:

- The node applies for a certificate and provides the subscriber, i.e. the CA, with all the necessary information.
- The CA checks whether everything is correct.
- If everything is correct, the CA adds further data to the node's data, such as their own identity, the hash procedure used, a unique serial number, etc.
- Finally, the CA signs the certificate with its own private key.

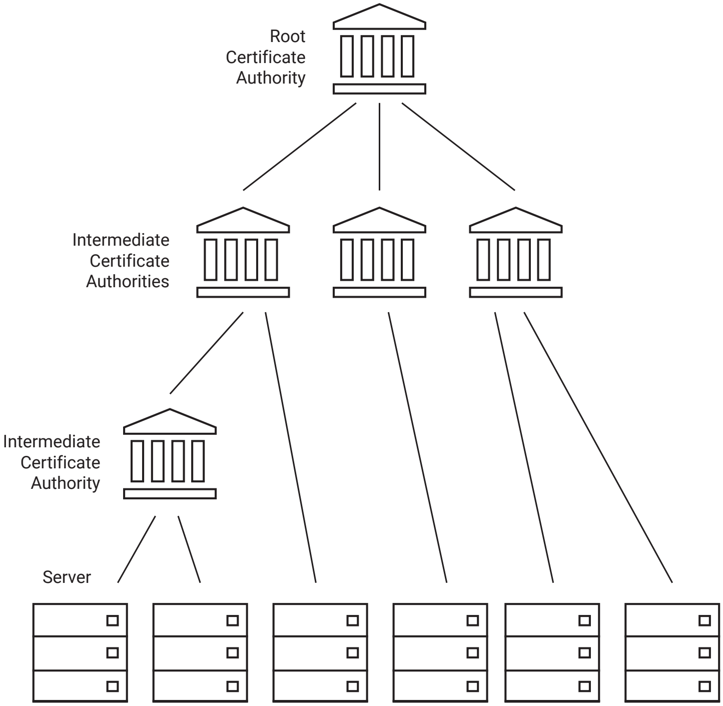
Ultimately, it is a matter of transferring the public keys of service providers or servers with a secure proof of origin to the client who wants to use such services. So it is about the distribution of the public keys.

Hierarchy of certification authorities

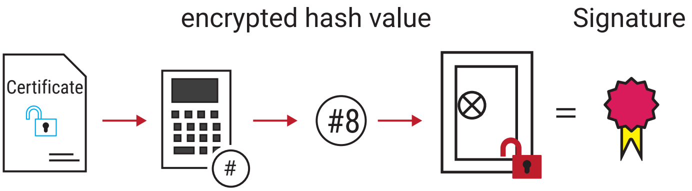
Due to the rapid expansion of the internet and the services it offers, the demand for signed certificates is enormous. A single trustworthy certification authority could not cope with this demand. But if there were numerous certification authorities, control and thus trustworthiness would quickly be lost.

That is why there is ultimately a limited number of trustworthy root certification authorities. These root certificate authorities place their trust in subordinate intermediate certification authorities.

These Intermediate CAs authorized by the CAs can in turn create certificates for servers (and clients). They can also authorize other intermediate CAs to issue certificates. This results in a tree-like CA hierarchy.

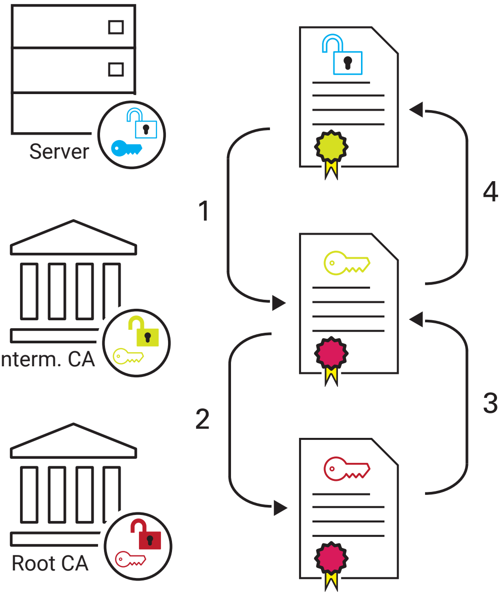

Root Certificate Authority
Intermediate Certificate Authorities
Intermediate Certificate Authority
Server

Remember: A signature is the hash value of the certificate content encrypted with the issuer's private key.



Each CA signs the issued certificates for the subordinate Intermediate CA using its private key. This results in certificate chains that lead back to the root CA level. Since there is no level above the root CA, the root CA issues its own self-signed certificate.

The subscriber (for example, a server) at the end of this chain must always provide all the certificates involved, right down to the root certificate.



If the client received a certificate that he doesn't know yet from addressed server, the client can trace the certificate chain back to the root certificate of the root CA (1 and 2).

If the root CA is known to be trustworthy, the chain can be checked for trustworthiness starting with the public key of the root CA and ending with the server certificate (3 and 4).

The server public key transmitted within the certificate can only be used securely for the pending communication once it has been checked successfully.

Die Public Key Infrastructure

In order to create, manage, distribute and, if necessary, revoke certificates, an infrastructure is required that enables this to be done securely in accordance with existing guidelines and standards.

A Public Key Infrastructure, PKI for short, consists of at least one root certification authority (Root CA) and various sub-certification authorities as required. A Registration Authority (RA) is also required. All certificates signed and issued within the PKI are listed and managed by the RA.

Certificates are checked for validity by the RA and can be revoked if necessary.

If, for example, a CAs private key were to fall into someone else's hands - which should not happen - the RA responsible would revoke all certificates issued by this CA.

Operating systems or applications such as browsers or mail clients must therefore regularly compare the deposited certificates with the RA via the PKI, e.g. by means of updates.

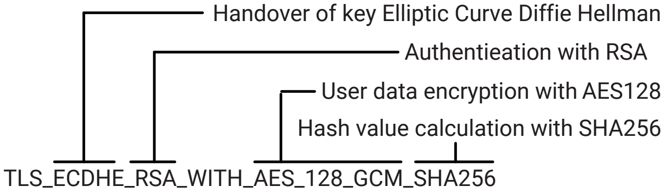
There are some recognized, trustworthy PKIs on the internet whose CAs/RAs issue and manage certificates. Usually this is done for a fee. For large corporations, organizations or public authorities, it may therefore be worthwhile to operate an in-house Public Key Infrastructure.

Cipher Suites

As we have learned so far, data transmission is carried out with symmetric and asymmetric encryption. There are different algorithms for both encryption techniques. This also applies to the hash value calculation.

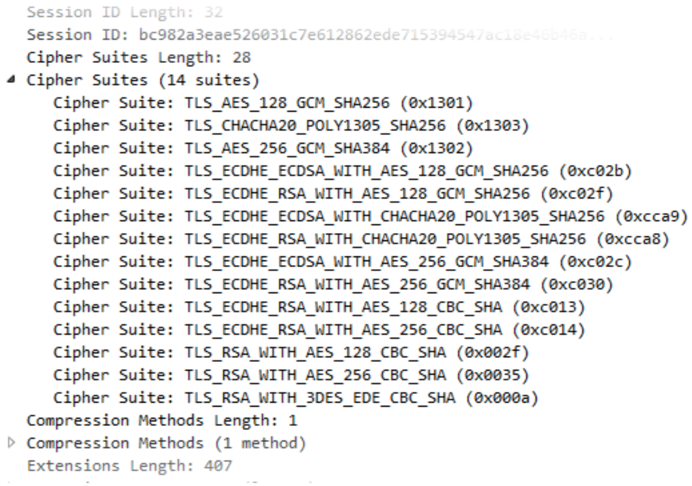
Before an encrypted data transmission begins, the communication partners must agree on a combination of the methods used.

These combinations are also called Cipher Suites and they contain the following information:


Handover of key Elliptic Curve Diffie Hellman
Authentieation with RSA
User data encryption with AES128
Hash value calculation with SHA256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

The Cipher Suites are not transmitted in the plain text shown here, but each usable combination has a unique 2-byte identification number. The one shown here for example is 0xC02F.

When the connection is established, the client sends a list of possible cipher suites to the server.



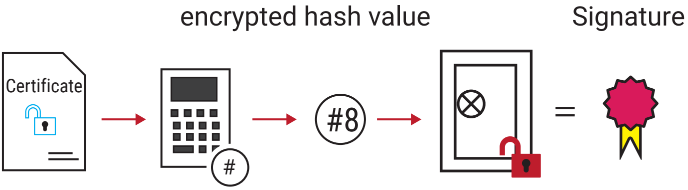
```
  Session ID Length: 32
  Session ID: bc982a3eae526031c7e612862ede715394547ac18e46b46a...
  Cipher Suites Length: 28
◢ Cipher Suites (14 suites)
     Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
     Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
     Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
     Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
     Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
     Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)
     Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)
     Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
     Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
     Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
     Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
     Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
     Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
     Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
  Compression Methods Length: 1
▷ Compression Methods (1 method)
  Extensions Length: 407
```

The server answers and tells the client which combination should be used.

Verifying Certificates

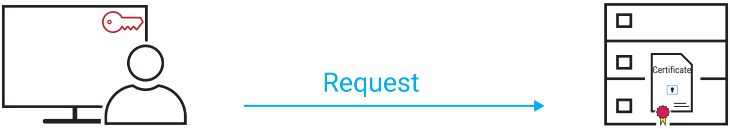
Up to this point we have explained very roughly how authentication via certificates works in principle. Since certificates are a very important part of security on the internet, it is worthwhile to take a closer look at how certificates are verified.

Just a reminder: For signing, the issuer, i.e. the CA, encrypts the hash value of the certificate content with the private key.

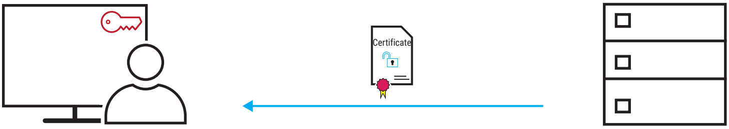


The subscriber receives the signed certificate, which consists of the unchanged certificate data and the signature, i.e. the encrypted hash value.

The communication partner (in this case the client), who wants a secure authentification from his counterpart, requests the signed certificate.



The certificate is immediately returned by the server.

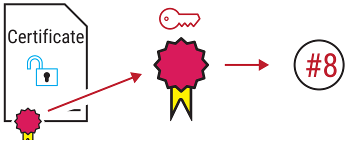


Thus the client receives the certificate data and the encrypted hash value matching the data.

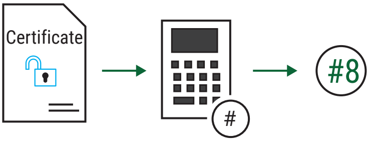
There is also a matching public key for the private key of the CA. For common standard applications such as web browsers, the public keys of the known CAs are integrated into the software (we will go into the administration of public keys in more detail later).

Public Key of the CA

Using the public key of the CA, the client decrypts the hash value encrypted by the CA.



In parallel, the client calculates the hash value of the certificate data, just like the CA did.



Now the client compares the decrypted hash value with the self-calculated one. If both values are equal, the authenticity of the certificate is confirmed

If the two values differ, it is not the original certificate and the client should immediately break off contact with the server.

Before the client starts the actual data exchange, it also checks the entries to ensure that the certificate has not expired, for example.

SSL/TLS

When it comes to encrypted data communication, the terms SSL and TLS quickly come up.

SSL (Secure Socket Layer) was developed by Netscape and was first introduced to the public in 1995 in version 2.0. Then one year later version 3.0 followed.

TLS (Transport Layer Security) is a further development of SSL and replaced it in 1999.

The working method of SSL and TLS is identical in many parts, which is why they are usually referred to as SSL/TLS.

SSL combines the security mechanisms described up to this point into a protocol sequence. Here once again we will briefly summarize how the different techniques are interlocked:

1. The client - e.g. a browser - establishes a TCP connection to the server
2. When the connection is established, the client informs the server which encryption methods it supports or wants to use. This is still done unencrypted.
3. The server answers - also still unencrypted - and announces which encryption method it has decided on.
4. In addition, the server sends its certificate.
5. The client checks the identity of the server on the basis of the certificate and thus ensures that it is connected to the correct communication partner.
6. When the identity of the server is confirmed, the client takes the public key of the server from the certificate.
7. The client generates a common key for a symmetrical connection and encrypts it with the server's public key.
8. the server decrypts the client's data transmission with its private key and thus receives the common key for symmetrically encrypted data exchange.
9. the data exchange encrypted in this way continues until one of the two communication partners terminates the connection.

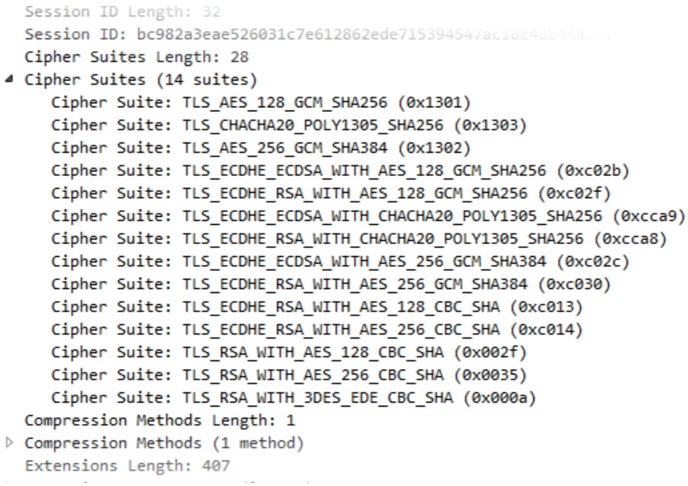
HTTPS - SSL/TLS in practice

Using the example of calling a web page in the browser via HTTPS, the process of an encrypted connection will be explained here once again.

The address is entered in the browser:

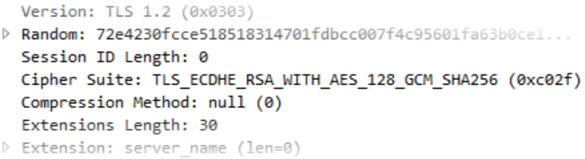
```
https://wut.de/
```

The browser establishes a connection to the WuT web server and sends a HTTPS request. In this request, it tells the server which Cipher Suites are supported.



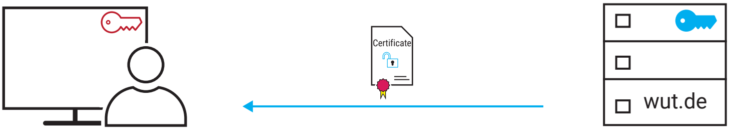
```
   Session ID Length: 32
   Session ID: bc982a3eae526031c7e612862ede715394547ac18e46b46a...
   Cipher Suites Length: 28
 ◢ Cipher Suites (14 suites)
      Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
      Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
      Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)
      Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
      Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
      Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
      Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
   Compression Methods Length: 1
 ▷ Compression Methods (1 method)
   Extensions Length: 407
```

The server selects one of the offered Cipher Suites and informs the browser about it.



```
   Version: TLS 1.2 (0x0303)
 ▷ Random: 72e4230fcce518518314701fdbcc007f4c95601fa63b0ce1...
   Session ID Length: 0
   Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
   Compression Method: null (0)
   Extensions Length: 30
 ▷ Extension: server_name (len=0)
```

The server then sends its certificate.

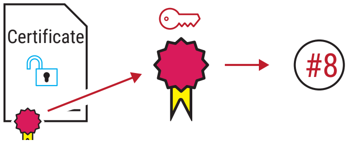


In this way the browser receives the certificate data and can first check whether or not the certificate has expired or has been revoked.

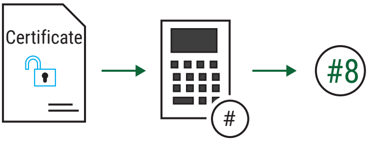
Using the public key of the CA

Public Key of the CA

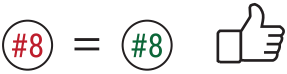
the browser decrypts the hash value encrypted by the CA and attached to the certificate.



In parallel, the browser calculates the hash value of the certificate data, just as the CA did.

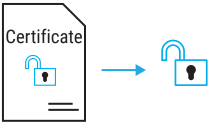


Now the browser compares the decrypted hash value with the self-calculated one. If the decrypted and the self-calculated hash value are equal, the authenticity of the certificate is confirmed.



This ensures that the browser is connected to the correct server.

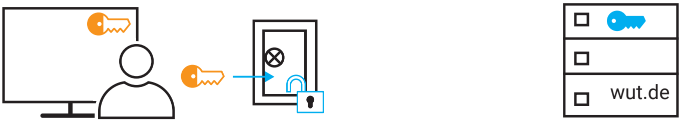
Now the browser can take the public key of the server from the certificate.



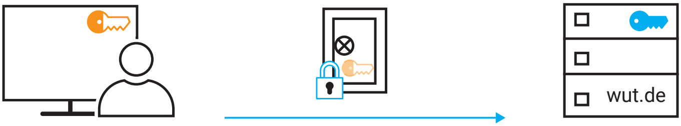
The browser generates a common key, which is used later in the browser session to exchange data that has been symmetrically encrypted.

🔑 shared
symmetric key
to encrypt and decrypt

The browser encrypts this common key with the public key of the wut.de server



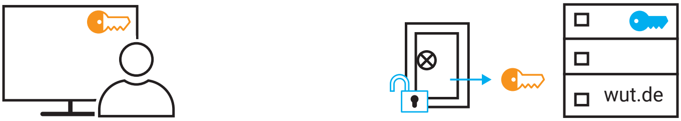
and sends it to the wut.de server.



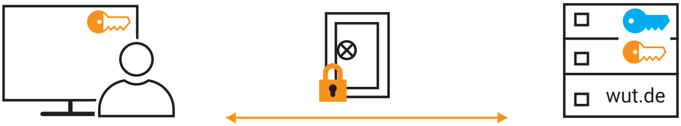
The wut.de server can decrypt the data transmission with the private key

🔑 Private Key
for decryption

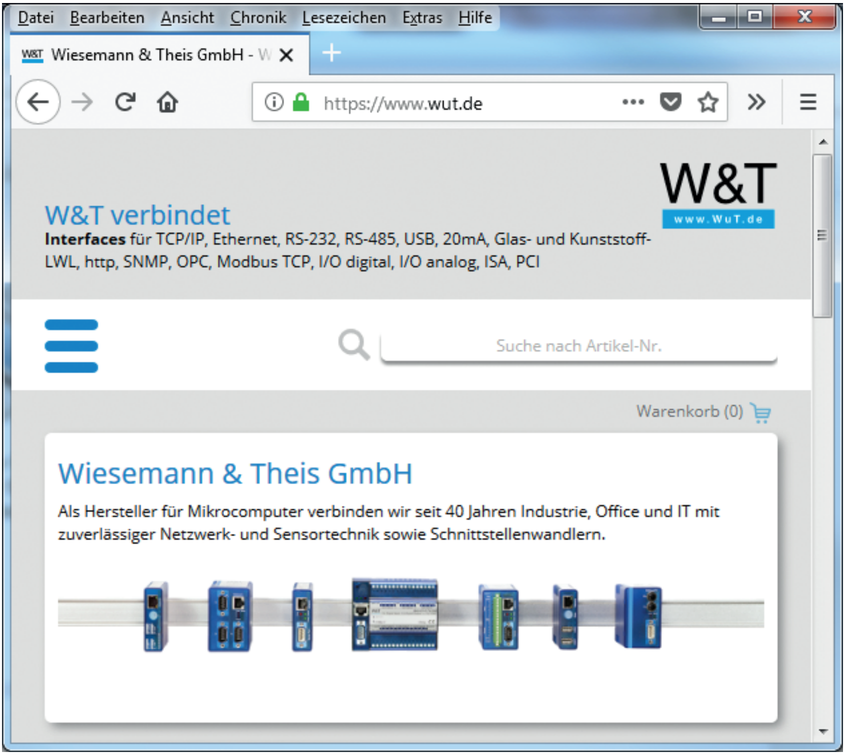
and thus gets the common key without others being able to spy out this key on the transmission path.



This is where the actual exchange of user data begins. Both the browser and the server can encrypt and decrypt the transmitted data with the same key.



This means that all data required to display the website is transmitted in encrypted form.



W&T verbindet

Interfaces für TCP/IP, Ethernet, RS-232, RS-485, USB, 20mA, Glas- und Kunststoff-LWL, http, SNMP, OPC, Modbus TCP, I/O digital, I/O analog, ISA, PCI

Wiesemann & Theis GmbH

Als Hersteller für Mikrocomputer verbinden wir seit 40 Jahren Industrie, Office und IT mit zuverlässiger Netzwerk- und Sensortechnik sowie Schnittstellenwandlern.

The padlock in the address bar shows the user that he is connected to the correct server.

VPN - Virtual Private Network

Basic facts

First of all, it should be said that the use and implementation of VPN allows various variants. If we were to cover all details concerning VPN that would provide enough material for a separate book and would go beyond the scope of this chapter. Therefor we will limit ourselves to introducing the global function and the most important basic terms of VPN.

VPN describes the technique of connecting confidential network parts at different locations via the internet, i.e. a public network. Typical examples for the use of VPN are

- **Cross-site network connections**
  Two or more company locations can be combined into a network infrastructure that can be used jointly.

- **Employees in the field service**
  For employees who work outside the company site, access to the company's internal network or parts of it can be set up. Access is then possible, for example, from the customer's network, via public hotspots, or the mobile phone network.

- **Home office**
  As for the employee in the field, a VPN access can be set up for an employee who works at home, allowing access to the company network from the home office.

- **Remote maintenance**
  Service technicians can connect to their customers' networks via VPN in order to carry out remote maintenance work, fault diagnosis, updates etc. on servers, controllers and machines.

Requirements for a VPN

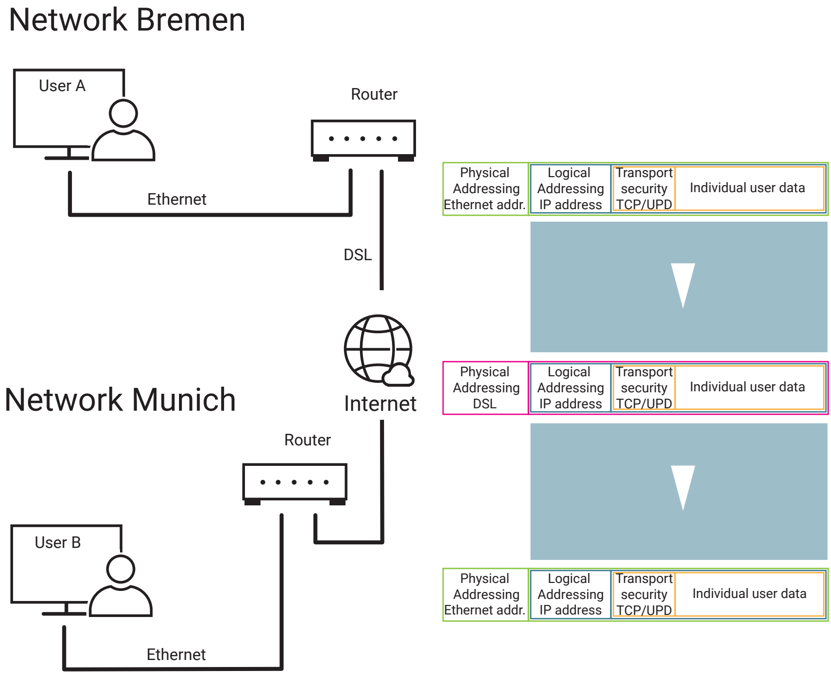
So, in the end it is always about getting access to a remote network.

In contrast to normal routing, VPN must also meet the typical requirements for data security:

- **Authentification**
  To access the remote network part, the access authorization must be proven.
- **Data integrity**
  When data is received, it must be ensured that it has not been changed during transport.
- **Data security / confidentiality**
  The transmitted data must be protected against falsification or interception by unauthorised third parties during transport.

Digression: normal routing

Remember: In normal routing, the source and destination networks have different Net IDs. The Net-IDs are part of the IP addresses and serve as routing information. The addresses within the IP data packet remain unchanged over the entire route.


Network Bremen
User A
Router
Ethernet
DSL
Internet
Network Munich
Router
User B
Ethernet
Physical Addressing Ethernet addr.
Logical Addressing IP address
Transport security TCP/UPD
Individual user data
Physical Addressing DSL
Logical Addressing IP address
Transport security TCP/UPD
Individual user data
Physical Addressing Ethernet addr.
Logical Addressing IP address
Transport security TCP/UPD
Individual user data

The physical address data, on the other hand, changes from subsection to subsection.

Within the local networks, addressing and data transport are carried out via Ethernet, and at internet level via DSL and other physical transmission methods. The IP packet remains unchanged over the entire distance to be bridged.

Data security

As long as the transmission paths are continuous, the data can be reliably routed from network A to network B, even with conventional routing.

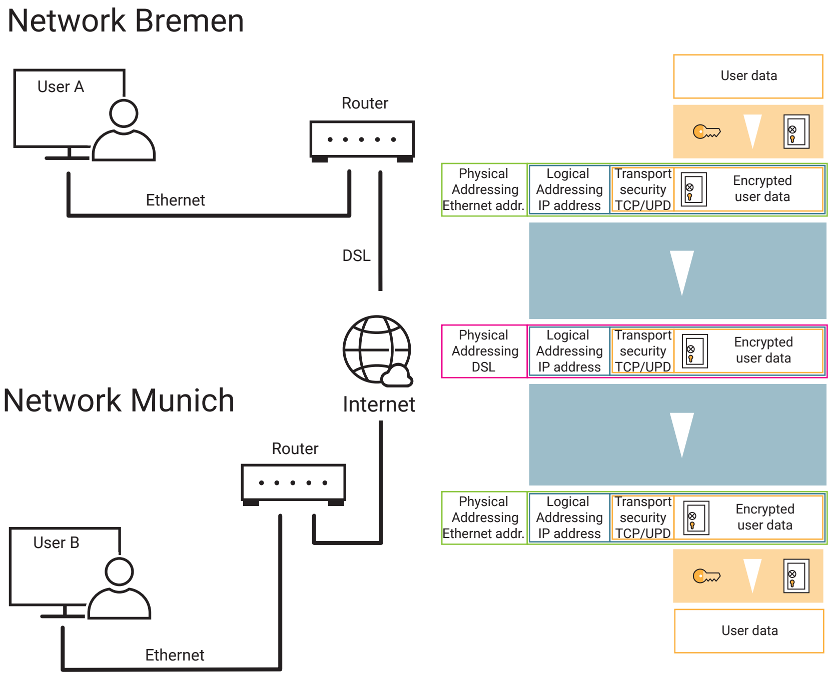
However, one disadvantage of normal network communication is that the data can be read by anyone who has physical access to the transmission paths. This means that a considerable security risk exists not only for bank data, for example.

Data encryption

As we have already learned in the previous chapter, one way to protect data from interception or manipulation is encryption.

Third parties, who do not know the key or keys used, cannot read or evaluate the encrypted data stream without further ado.

Both sides must know the key used, or an asynchronous encryption or hybrid method must be used.


Network Bremen
User A
Router
Ethernet
DSL
Internet
Network Munich
Router
User B
Ethernet
User data
Physical Addressing Ethernet addr.
Logical Addressing IP address
Transport security TCP/UPD
Encrypted user data
Physical Addressing DSL
Logical Addressing IP address
Transport security TCP/UPD
Encrypted user data
Physical Addressing Ethernet addr.
Logical Addressing IP address
Transport security TCP/UPD
Encrypted user data
User data

However, the IP and TCP addressing parameters are readable. Third parties who want to access external data or a foreign network can at least see from this data where the target network is vulnerable from the inside.

VPN instead of normal routing

As mentioned at the beginning, VPN allows two network parts at different locations to be connected via the internet or a public network.

Even though VPN is based on the normal IP network mechanisms, there are significant differences behind the scenes.

VPN - Possible topologies

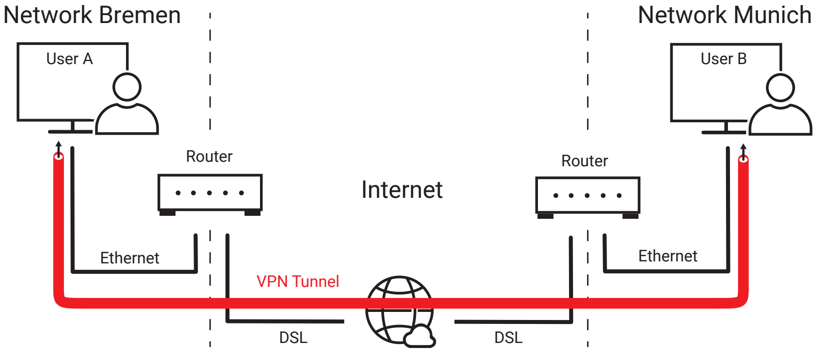
There are three basic topologies for VPN solutions:

- End-to-End
- Site-to-Site
- End-to-Site

Which variant is used ultimately depends on how the VPN connection is to be used.

VPN - End-to-End

In end-to-end solutions, two network terminals are connected to each other via a public network - e.g. the internet - in such a way that they can exchange network packets with each other without restriction. The transmission route through the public network is also known as a tunnel, as the data traffic between the terminals is separated from the rest of the network traffic.

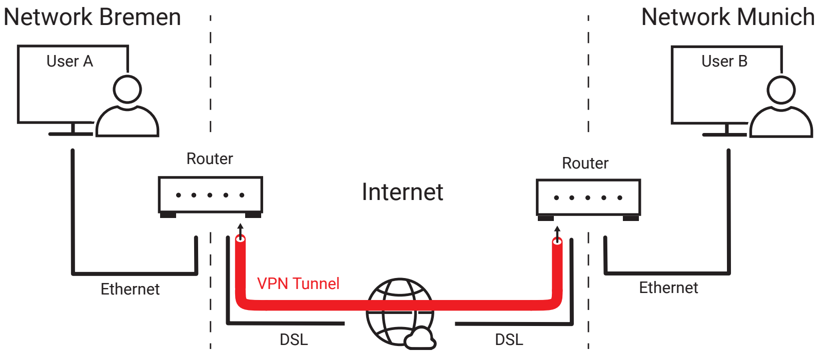


An example of an end-to-end VPN connection is a maintenance access, that a service technician can use from his office to analyze errors on a customer system, for example.

However, for the whole thing to work, special VPN software must be installed on both PCs. Furthermore, each PC must be specially configured for VPN access.

VPN - Site-to-Site

With the VPN site-to-site technology, two individual networks are connected, e.g. via the internet.



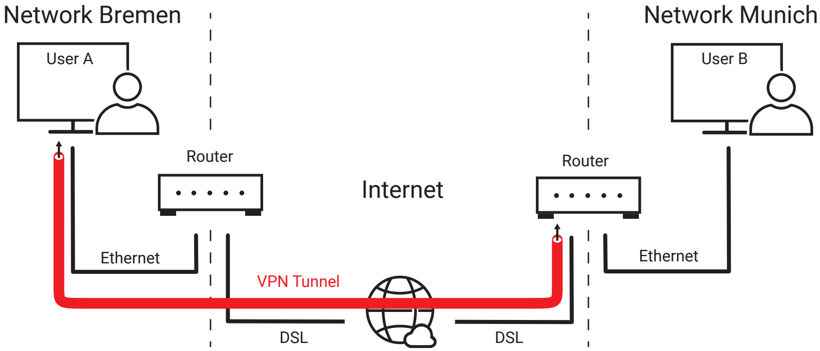
The VPN tunnel is established between two special VPN routers. The entire VPN configuration takes place in the routers.

The individual participants in the network do not require any special software and do not have to be configured separately.

Site-to-site solutions are mainly used to connect different company locations.

VPN - End-to-Site

The end-to-site solution provides individual end devices or PCs with access to an entire network at the remote location.



This solution is ideal for connecting home office workstations. The employee can use the entire infrastructure of the company network from home.

VPN-Protocols

For the technical implementation of VPN, several protocols come into question in practice:

- PPTP - Point-to-Point Tunneling Protocol
- IPsec - IP Security Protocol
- L2TP - Layer 2 Tunneling Protocol
- OpenVPN
- WireGuard

Which protocol is used depends on the VPN topology and the hardware and software used.

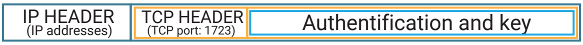
PPTP - Point-to-Point Tunneling Protocol

Originally PPTP was developed by Microsoft and 3COM to allow remote PCs to access central servers via a dial-up line. Since PPTP is implemented in Windows operating systems by default, it is still widely used. However, the encryption of PPTP is no longer considered secure.

The technical basis of PPTP is the PPP protocol (see chapter: Transmission Protocols) which, among other things, has been extended by data encryption and additional authentication.

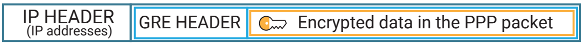
Due to the PPP implementation, PPTP has the advantage of being able to transmit other protocols besides IP, such as IPX (formerly used by Novell and Windows).

PPTP works in two steps: First, authentification and key data are exchanged via a control connection on TCP port 1723.

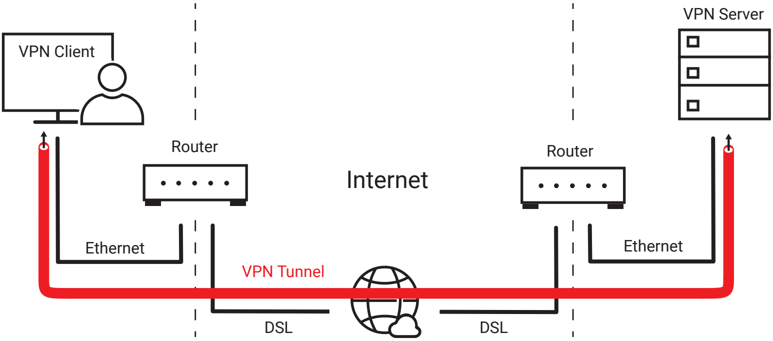


The PPP data is then exchanged encapsulated in the GRE protocol. The GRE (Generic Route Encapsulation) encapsulation is, metaphorically speaking, the tunnel through which the PPP data is transported.

GRE has the character of a transport protocol that works on the same level as, for example, TCP and is embedded directly into an IP packet.



PPTP works according to the client/server method. The VPN client logs on to a VPN server when the control connection is established. Therefore only end-to-end VPN solutions are possible.



IPsec - Internet Security Protocol

IPsec was specially designed for the secure data transmission of IP data traffic via public networks or the internet. As a security protocol, IPsec contains various functions to meet the usual requirements of data security.

SA - Security Association

The communication partners have the possibility to negotiate which procedure or standard should be used for the following points:

- Authentification
- Data integrity
- Data security
- Key exchange

The resulting parameter set is called Security Association.

SAD - Security Association Database

The parameter sets are stored in the Security Association Database, SAD for short. When VPN connections to more than one target are operated, different parameter sets for each individual target can be managed in a kind of list. To implement the above mentioned security mechanisms in the protocol, IPsec offers two security protocol variants which can be used individually or in combination.

AH - Authentication Header

In case of Authentication Header, AH for short, only authentication and data integrity are secured. This ensures that you are dealing with the desired communication partner and that the data has not been changed during transmission. However, the transmission is not encrypted, so that third parties may be able to read the contents.

ESP - Encapsulating Security Protocol

The Encapsulating Security Protocol, ESP for short, also ensures authentication and data integrity, but in addition the data is encrypted. This naturally requires corresponding keys on both sides of the VPN connection.

IKE(v2) - Internet Key Exchange (Version 2)

IPsec uses the IKE protocol to make the required keys and other parameters accessible to both sides. IKE uses UDP port 500, and the original IKE has now been replaced by IKEv2.

IKE works in two phases:

- **Phase 1**
  In the first phase, authentication is secured - in other words, it is ensured that communication takes place with the actually desired partner. A secure communication channel is then established for the second phase.

- **Phase 2**
  In the second step, agreement on the security mechanisms used is reached via the secured channel.

Besides the different security mechanisms, there are also two different transport models:

- **IPsec Transportation**
  The data transport is carried out via normal routing, whereby within the public network all data except the IP headers are protected against external access.

- **IPsec Tunneling**
  The network group created by VPN tunneling presents itself to the network users as a local network. The entire data stream including IP header is protected.

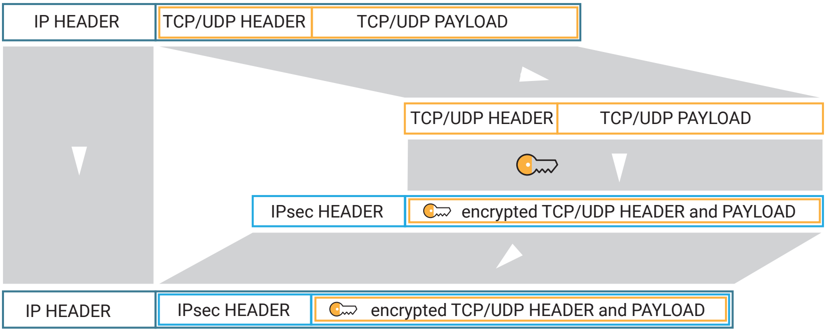
IPsec Transportation

The IPsec transport mode is preferred for end-to-end VPN solutions. In order for this to work, software is handle the IPsec procedure must be installed on the PCs involved.

In order to receive the data securely over the public network, the IPsec driver extracts all contents that are above the IP part in terms of protocol from the sent TCP/IP packets.

The entire TCP or UDP part - i.e. header and payload (user data) - are encrypted together and packed into an IPsec frame.

The IPsec frame is then built into an IP packet, whereby the original IP address information is retained.

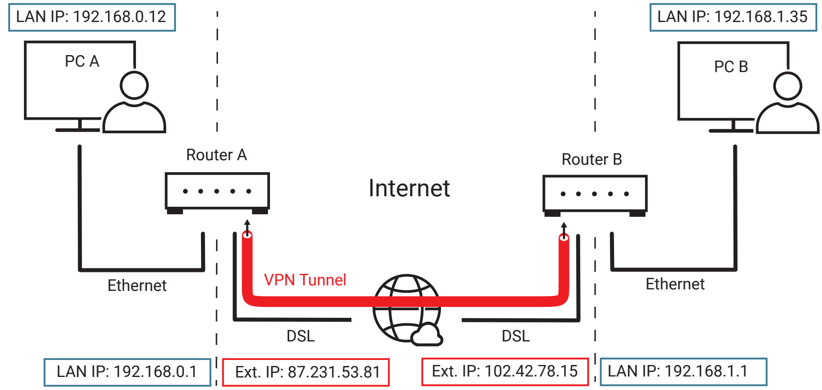


The data transport is thus handled with normal routing, whereby the transported data and port information is protected.

IPsec tunneling

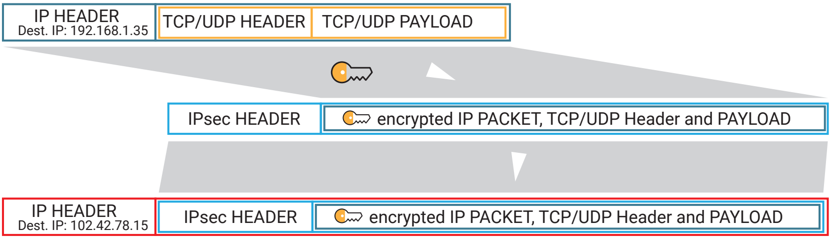
As mentioned at the beginning, IPsec tunneling can be used to connect two sub-networks at different locations via the internet as if data were being exchanged between two local sub-networks (site-to-site solution).

The handling of IPsec is done by special routers. The advantage of IPsec tunneling over IPsec transport is, among other things, the relief of the participating end devices. Special drivers are not necessary.



If PC A sends a data packet to PC B, it is first received by router A.

Router A encrypts the entire IP packet portion as it is and packs it into an IPsec frame. The IPsec frame is then built into a new IP data packet addressed to router B.



Router B decrypts the original IP packet and sends it to PC B.

For the PCs, the data transmission appears as if the data traffic was routed normally.

Routing within the internet, however, takes place exclusively between the two VPN routers.

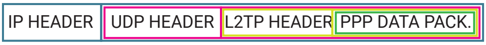
If necessary, any terminal in network A can exchange data with any terminal in network B, securely and as if the remote terminal were in the same local network.

With the various available transport and security modes, IPsec can be used in a highly flexible manner, but involves a high configuration effort.

L2TP - Layer 2 Tunneling Protocol

L2TP is a pure tunneling protocol.

For data transmission, L2TP uses the PPP protocol just like PPTP. The PPP data is provided with an L2TP header and embedded in a UDP packet.

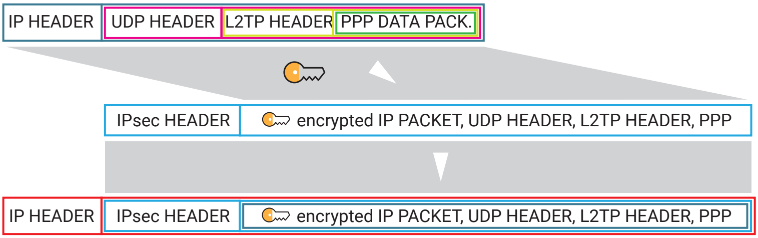


L2TP performs the following tasks:

- Mounting and dismounting of a data tunnel
- Checking if data have reached their recipient correctly
- Numbering of the data packets in order to put the data in the correct order at the recipient

However, L2TP works completely unencrypted. Unauthorized third parties who have access to the transmission paths could read all information unhindered. Therefore L2TP alone is not suitable for the realization of a VPN tunnel.

To gain the necessary security, L2TP is usually used together with IPsec.


IP HEADER
UDP HEADER
L2TP HEADER
PPP DATA PACK.
IPsec HEADER
encrypted IP PACKET, UDP HEADER, L2TP HEADER, PPP
IP HEADER
IPsec HEADER
encrypted IP PACKET, UDP HEADER, L2TP HEADER, PPP

Now one could of course ask: If L2TP alone is not secure anyway, why not use IPsec?

- Firstly, there are applications where data has to be tunneled within a confidential network - here L2PT offers everything that is needed.
- Secondly, IPsec can only tunnel IP packets. L2TP can also transport other types of packets because of the PPP protocol that is used - using IPsec these can also be encrypted.

OpenVPN

OpenVPN is licensed as free software under GNU GPL (General Public License).

The variety of functions and the abundance of configuration possibilities of OpenVPN is very large, so that we only want to summarize the features and possibilities here:

- OpenVPN is based on the OpenSSL library and can therefore use all encryption, authentication and certification options of SSL/TLS.
- Either UDP or TCP can be used as the basic protocol. If TCP is used, the communication is done via port 443, i.e. the HTTPS port - This means that OpenVPN can overcome most firewalls without any problems.
- It runs on almost all known operating systems and many standardized hardware platforms (and can therefore be used in routers, embedded systems and smartphones in addition to PCs and servers).
- It can be combined with IPsec.
- All three VPN topologies (end-to-end, site-to-site and end-to-site) are supported.
- Even very large network segments with more than 1,000 remote accesses can be set up with OpenVPN.

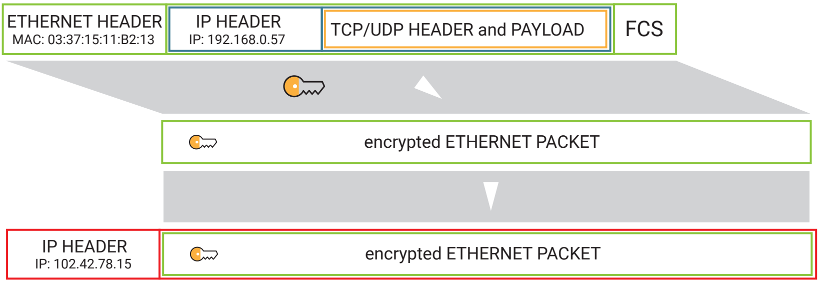
A special feature of OpenVPN is that besides normal routing, bridging is also supported.

Routing

As a reminder: During routing, the IP address and subnet mask are used to determine whether communication takes place in the local network or via which route the data is further transmitted to which target network. Of course, this only works at the IP level.

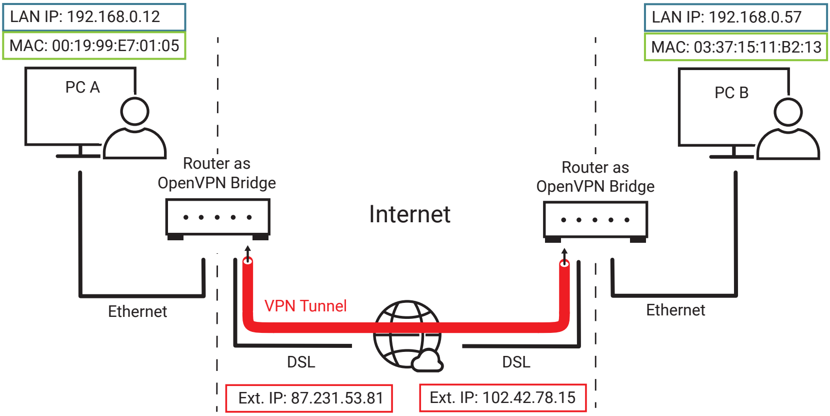
Bridging

When it comes to bridging, the complete Ethernet data packet is transferred from one subnetwork to another. To do this, the entire packet is first encrypted and then embedded in an IP packet that is addressed to the external IP address of the second bridge.



For IP data traffic, this has the advantage that both subnetworks can be in the same IP address range.

For the individual network node the network of subnets acts like a single local network.



Furthermore, the transmission is not limited to IP data packets. Other protocols such as IPX can also be transported.

Even if it sounds paradoxical at first - OpenVPN is considered to be particularly secure because all source codes are freely accessible. This ensures that programming errors or hidden backdoors can be detected very quickly.

WireGuard

WireGuard is the youngest of the VPN technologies and also an OpenSource project.

During the development of WireGuard, the following criteria were given priority:

- carefully planned approach
- easy setup and handling
- top performance
- use of current encryption and security techniques
- small amount of source code

WireGuard uses UDP for transmission, whereby the port is freely selectable.

At the time this edition went to press, the development of WireGuard had not yet been completed. Further details about the WireGuard project can be found at

https://www.wireguard.com..

Access the internet

A decisive limitation of today's standard Ethernet technology is the maximum distance of 100m. Although longer distances can be achieved with the help of appropriate components such as hubs, switches and routers. However, even when such components are used the reach of an Ethernet network is limited to the premises of a company or home of a private user.

If, for example, a connection to the internet has to be established (remote data transmission), several kilometers often have to be bridged. With few exceptions, internet access is therefore connected via the public telephone, cable TV or mobile phone network. In rare cases, satellite radio links are also used to connect to the internet.

Historical internet accesses

The telephone network in particular has undergone major technical changes in recent years. Originally designed exclusively for voice transmission, but since the end of the 1990s more and more data services have been transmitted via the existing technology and infrastructure. Both services have now grown together as ALL-IP connections and use TCP/IP as their basic technology.

As a result, some access technologies such as analog modems and ISDN are losing their relevance. However, since these technologies provide a basic knowledge of the transmission technology that is still required today, we would like to go into this briefly.

Analog modems

Modem stands for modulator-demodulator. Access via analog modems was the original way of accessing the internet and is no longer used today, at least not for access to the public network. However, for in-house applications, e.g. for bridging larger distances on a company site, analog modems are still used.

A modem is connected between the terminal device, usually a PC, and the telephone connection or a telephone line. The serial interface (COM port / RS232) or USB is usually used as the interface between the PC and the modem. As an alternative to the external modems, there are PC plug-in cards which handle the modem functions within the PC.

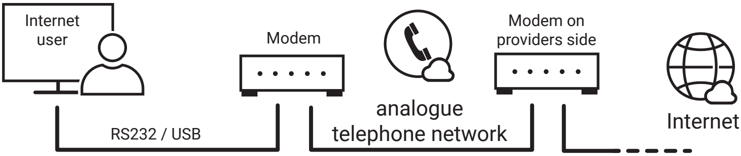
If the public telephone network was used for transmission, a dial-up connection to the internet provider had to be established first. This task was also performed by the modem.

For transmission, the digital information was modulated onto a carrier frequency. At this point we do not want to go into detail about the modulation methods used, but only to give an exemplary explanation of this technique.

The carrier frequency can be imagined like a certain audible tone from the frequency range of speech (300 Hz - 3,400 Hz).

The data stream to be transmitted is divided into blocks of a few bits. Depending on which bit pattern is present, the sound is changed in a way specified for this bit pattern.

At the other end of the connection path, a second modem takes over the reverse task in reverse (demodulation). A data stream is recovered from the received tones.



Historical internet access via telephone network and modem

Due to the limited frequency range of analog telephone connections or the lines have been laid, the maximum data transmission rate is 33Kbit/s from the subscriber to the exchange (upstream). From the exchange to the subscriber (downstream) a maximum of 56Kbit/s is possible.

In addition to the low transmission rate, a major disadvantage of dial-up via analog telephone connections was the fact that it was not possible to use telephone at the same time.

ISDN - Integrated Services Digital Network

The introduction of All-IP connections has replaced ISDN technology, so that ISDN no longer exists in the German telephone network.

The main difference between ISDN and the analog telephone connection was that in case of ISDN even analog voice data was converted into digital switching data at the subscriber’s location.

From the subscriber to the exchange, therefore, only digital data was exchanged in the form of ISDN network packets.

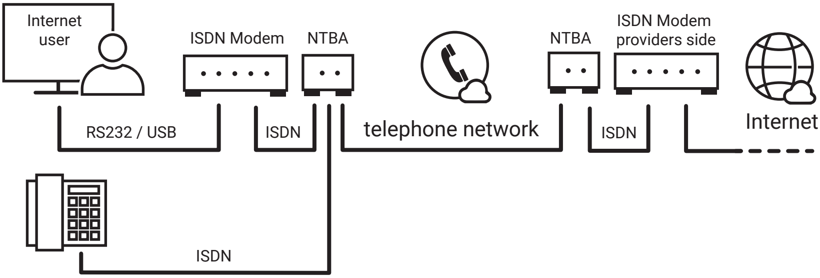
ISDN stands for Integrated Services Digital Network,

In addition to the transmission of speech, ISDN allowed the exchange of digital data, e.g. for fax and dial-up.

A modulation of dial-up data was not necessary when using ISDN in the true sense of the word. Instead, the data to be transmitted was packaged and sent in ISDN packets, and here too a dial-up connection was required initially.

Nevertheless, the external ISDN <-> dial-up data converters were commonly referred to as ISDN modems.

Between the ISDN modem and the telephone network, the NTBA (Network Termination for ISDN Basic rate Access) physically prepared the ISDN data so that it could be transmitted to the switching office. The interface between the ISDN terminals and the NTBA was called the S0 bus.



ISDN provided the subscriber with two channels (areas in the ISDN package), which could also be used for different services, e.g. telephony and dial-up.

Per channel 64Kbit/s were transmitted. If both channels were used in parallel (channel bundling), the transfer rate increased to 128Kbit/s.

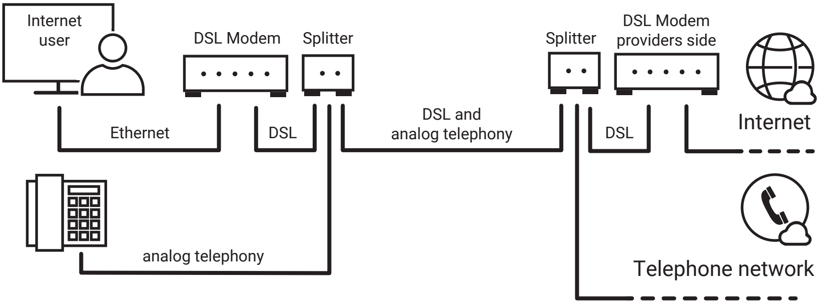
Current internet accesses

DSL - Digital Subscriber Line

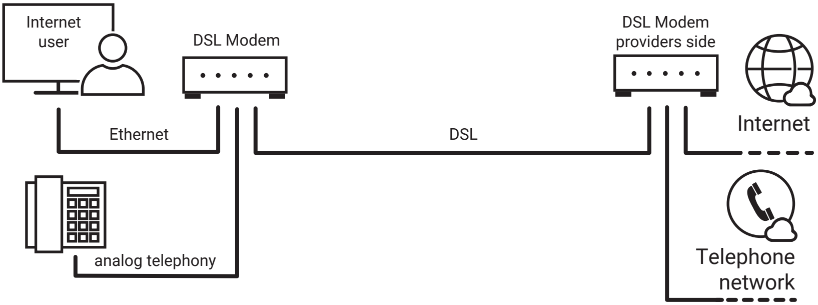
For a long time the Digital Subscriber Line was most attractive way to connect to the internet.

Analog connections work on the cable with frequencies up to max. 3.5 kHz. In case of ISDN the upper limit is approx. 40 kHz. DSL only uses frequencies above 40 kHz to approx. 1 MHz.

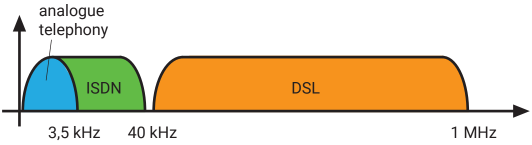
This meant that DSL could be operated in parallel with analog or ISDN connections over the same cable. At the location of the subscriber line, a splitter (a crossover network) was used to separate the DSL signal from the telephone signals.



When using today's standard all-IP technology, telephony and data are first separated in the router.



The transmission of DSL data is similar to that of an analog modem, except that several, significantly higher carrier frequencies are used simultaneously.



DSL is available in different variants:

ADSL - Asymmetric Digital Subscriber Line
ADSL accesses are mostly used by private customers and allow transfer rates of max. 25Mbit/s when downloading. As the upstream speed is only about one eighth of the downstream speed, this is also called asymmetric data transmission.

SDSL- Symmetric Digital Subscriber Line
SDSL uses the same transmission speed of max. 4Mbit/s in both directions. SDSL accesses are the method preferred by commercial customers - e.g. to connect two company locations with each other in terms of network technology.

VDSL - Very Highspeed Digital Subscriber Line
As more and more services such as television or telephony use the internet as a transmission path, the demand for very fast internet access is growing. VDSL works in a similar way , but with significantly higher transmission rates of over 200Mbit/s.

The following applies to all DSL standards: The greater the distance to the switching office, the lower the possible transmission speed.

Due to the high transmission speed, DSL modems exchange data with the PC directly via Ethernet. A common variant is an Ethernet router with integrated DSL modem.

Cable modem

Internet access via a cable modem has become a real alternative to a DSL connection. Access is provided via the cable television network. In the 1980s the cable television network was set up to distribute television and radio channels and was only intended to transport signals from the provider to the customer. When the network operators recognized the increasing demand for internet access, the system was retrofitted with necessary return channel from customer to provider

Physically, the existing networks are set up with coaxial cables. Fiber optic cables are used for network expansions and new networks.

The cable modem or a specially equipped router provides the connection between the cable television network and the local network.

The possible transmission speed is 32Mbit/s and more.

Internet access via the mobile phone network

An alternative to the previous fixed network variants is to connect to the internet via mobile networks.

A detailed description of mobile communications technology would go beyond the scope of this book. That is why we only want to give a very superficial overview at this point.

Mobile phone standards are just entering the fifth generation and although the 5G network is just starting to be built in Germany, 5G is already on everyone's lips.

First of all, here is an overview of the mobile phone generations:

- 1G
  The first generation started in 1958 with the A-Net. It was almost exclusively about telephony in cars. Although the required technology almost filled the whole trunk of a car, the subscriber could not dial for himself. Connections were handled manually via the analog A-network. Data transmission was not yet an issue at that time. From 1972, subscribers were able to dial for themselves using the B-network. In 1986, the C-network was established and the first portable mobile phones were available.

- 2G
  With the arrival of the D-Network in 1992, a switch from analog voice transmission to digital technology was made. One year later, the E-network was set up. Both networks use GSM (Global System for Mobile Communication) as the technical standard and now also allow data transmission - initially in the form of SMS text messages and very slowly at 9.6 Kbits/s.

- 2.5G
  GPRS (General Packet Radio Service) made the world of mobile telephony internet-capable in 2001. This was still very slow at max. 54 Kbits/s.

- 2.75G
  When EDGE (Enhanced Data Rates for GSM Evolution) arrived in 2006, the possible transmission speed in the GSM network was increased again to 150 Kbits/s by using new modulation and compression procedures.

- 3G
  In 2004, UMTS (Universal Mobile Telecommunications System) was introduced as the new mobile communications standard. It required the construction of a new, more closely meshed network and new terminal equipment. Due to the significantly higher bandwidth (approx. 380 Kbits/s) of UMTS, this also heralded the triumphal march of mobile internet use.

- 3.5G
  In 2006, HSPA (High Speed Downlink Packet Access), an extension of UMTS, brought about a further significant speed boost of up to 42 Mbits/s.

- 4G
  In 2010, LTE (Long Term Evolution) was introduced as the current mobile communications standard. LTE is based on UMTS technology. While initially transmission rates of up to 50 Mbits/s were possible, frequency bundling and other techniques (LTE Advanced) have enabled speeds of up to 400-500 Mbits/s to be achieved.

- 5G
  As mentioned above, the 5G network is currently under construction and is expected to be at least partially operational by 2020. Transfer rates of up to 20 gigabits/s are promised. Almost more important, however, are the very short latency times (response times of the network services) that 5G is supposed to provide. This means that 5G comes close to real-time behavior, as required for industrial applications and autonomous driving, for example.

Here is an overview of the different standards:

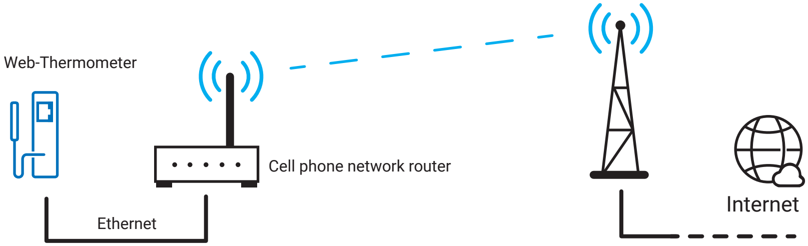
Generation	Launch year	Standard	max. Data rate
2G	1992	GSM	9,6 Kbits/s
2.5G	2001	GPRS	54 Kbits/s
2.75G	2006	EDGE	150 Kbits/s
3G	2004	UMTS	380 Kbits/s
3.5G	2006	HSPA	42 Mbits/s
4G	2010	LTE	500 Mbits/s
5G	2020	5G	20 Gbits/s

The specified speeds are maximum transfer rates. Which speed is actually achieved depends, among other things, on the distance to the next radio mast and how many participants exchange data in the same radio cell.

Further explanations of the technology behind the described mobile radio standards can be found in the network ABC.

Technical requirements

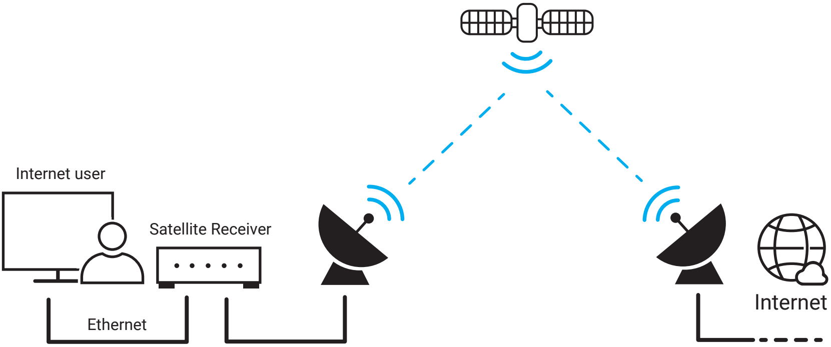
To connect to the internet as a user via the mobile phone network, you need either a smartphone or tablet, a surf USB stick or a mobile router.



To connect any Ethernet end devices to the internet via mobile radio, routers with the corresponding mobile access can be used.

Internet access via satellite

For very remote locations there is the possibility to get internet access via a satellite connection. The hardware expenditure for such a connection is much higher than for the usual access methods, because a parabolic antenna and a special satellite receiver are required.



Internet connections via satellite allow transmission rates of up to 30Mbits/s downstream, i.e. data traffic to the user. The upstream speed is significantly slower. With IoT applications one should keep in mind that the latency times are significantly longer than with all other internet connections.

The browser as user interface

First of all, this chapter is not intended to be a tutorial for creating web pages or web applications - the examples given only give an overview of the possibilities and technical background.

In the first 20 years of its existence, the use of the internet was of little interest to the ordinary user. A small group of insiders, by today's standards, had to type in cryptic command lines to exchange information.

Today, life without the internet is hardly conceivable. Banking transactions, online orders, holiday bookings, partner searches - today almost everything can be done via the internet. In 2019, every adult in Germany spent an average of three hours a day on the internet - young people spent as long as six hours.

With smartphones, tablets, notebooks and networked cars, the internet is with you virtually all the time.

The breakthrough for this internet triumph came with the introduction of the browser as a visualization tool that can be handled by normal users.

WWW - World Wide Web

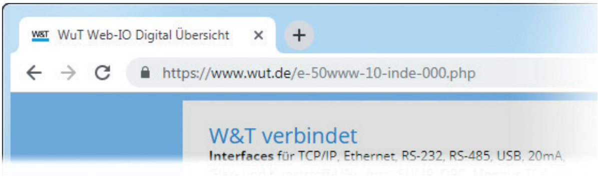
Nowadays, everyone uses the browser as a matter of course, without worrying about the technology behind it.

As early as 1994, the World Wide Web Consortium, or W3C for short, was founded - an organization whose mission is to create uniform standards for Web technologies worldwide.

Again as a reminder: The browser is a client application and it establishes a connection to the desired web server if required. The transfer of data is handled via the HTTP or HTTPS protocol (for details, see the chapter Web Protocols).

URL - Uniform Resource Locator

The URL is the address that you enter in the browser and which specifies where the desired content is retrieved.



https://www.wut.de/e-50www-10-inde-000.php

W&T verbindet

Interfaces für TCP/IP, Ethernet, RS-232, RS-485, USB, 20mA,

In addition to the actual address of the web server, the URL contains further information and parameters:

```
protocol://hostname [:tcp port] [/pathname][/filename][?further parameters]
```

`protocol`

For calling a web page usually HTTP or HTTPS

In the past, FTP or TELNET protocols were also supported depending on the browser and operating system. When specifying the protocol, there is no difference between upper and lower case.

`hostname`

Domain name of the server or IP address. Again, no distinction is made between upper and lower case.

`:tcp port`

Specifying the TCP port is optional and only needs to be done if the standard ports for HTTP (80) or HTTPS (443) are not used.

`/pathname`

Similar to a PC, a web server also has a directory structure. If the desired content is located in a subdirectory, the path leading there is specified with a preceding slash. The path is case-sensitive.

`/filename`

The name of the file to be called up can be entered here. If no file name is specified, the web server uses the file “index.html” or “index.php”. The file name is also case-sensitive.

`?further parameters`

Separated with a question mark further parameters can be specified within the URL. Several parameters are separated by “&”. Whether upper/lower case is relevant depends on the programming of the web page that has been called.

Example - URL for the privacy policy of Wiesemann & Theis:

`https://www.wut.de/e-wwwww-ds-rdde-000.php?Reference=datenschutz`

HTML – Hypertext Markup Language

After calling the URL, the browser receives the corresponding web page as a HTML file from the web server. In HTML format, the browser is told which contents are to be displayed and how.

HTML files have the name extension .htm or .html.

HTML is a markup language that consists of keywords - also called tags - and the content to be displayed. The tags indicate the way in which the following text is to be displayed. For example, font size, font type and alignment can be specified. Content can be displayed in tables or in the form of a numerical enumeration, the color of text and background can be specified, etc. The browser interprets these specifications and displays them accordingly.

HTML-Tags

There is a strict schema for HTML tags:

- Single tags are enclosed in angle brackets
  `<HTML-Tag>`.
- The actual tag can be extended by specifying attributes.
  `<HTML-Tag Attribut="xy">`
- Predominantly, the use of tags in pairs determines the beginning and end of their validity range; the defined properties then apply to everything between the tags. The closing tag repeats the opening tag with a preceding forward slash.
  Example`<title>Willkommen</title>`
- HTML tags are not case-sensitive
  `<HTML>` is equivalent to `<html>`.

Basic structure of a HTML file

In the first line, the `<!DOCTYPE ...>` tag indicates that this is an HTML file.

The actual content is introduced with `<html>` and ends with `</html>`. In the further structure of a page, a distinction is made between head and body.

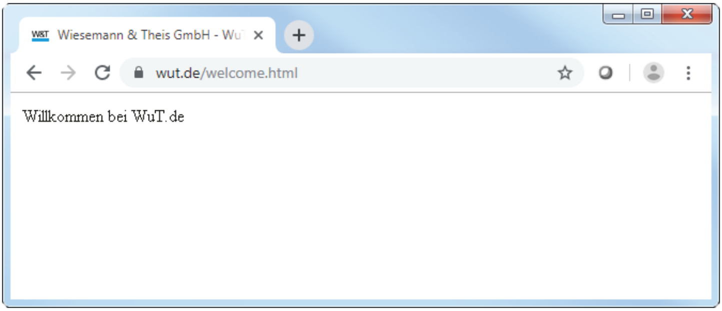
All information in the header remains invisible to the viewer and contains properties of the page that do not directly affect the display. The only exception is the title that is displayed in the title bar of the browser window. The header information is located between the tags `<head>` and `</head>`.

The head is followed by the side body, which is introduced with the `<body>` tag. The body of the HTML page contains all information concerning the actual content of the page and its presentation. The end of the body is marked with the `</body>` tag.

Here is a simple example:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Wiesemann &amp; Theis GmbH - WuT</title>
  </head>
  <body>
    Willkommen bei WuT.de
  </body>
</html>
```

In the browser it looks like this:



Willkommen bei WuT.de

In addition to texts, graphics can also be integrated with the help of HTML. Even multimedia content such as music, speech or film sequences can be integrated using HTML. The HTML document itself only transports text content. For every other element to be displayed, HTML is used to specify from where it can be loaded, where it should appear on the screen and in what size it should be displayed.

A good overview of HTML and explanations of all available tags can be found at https://wiki.selfhtml.org.

Hyperlinks

Probably the most important feature of HTML is the use of hyperlinks. Texts and other elements of a Web page can be provided with a hyperlink, that is, a URL reference to another Web page. If the user clicks on such a linked element, he is redirected to the linked web page.

We extend the HTML code with a hyperlink:

```
<body>
  Willkommen bei <a href="https://www.wut.de">WuT.de</a>
</body>
```

With a mouse click on “WuT.de” we are now directed to the homepage of W&T.

The path attribute of the tag <a href=”path specification”> can contain the path specification either in absolute or relative form.

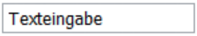
- Absolute: The complete URL to which the hyperlink should point is specified.
- Relative: Only the name of the file to be accessed is specified. The file is then searched for in the same directory where the current HTML file is located.

Forms

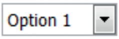
If it is required that the user can also send information to the web server HTML provides the option of including forms.

Forms contain elements that allow the user to enter text or mark something in a selection, for example. Here are the most important form elements:

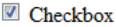
Text fields


Texteingabe

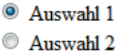
Text selection boxes


Option 1

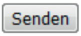
Check boxes


☑ Checkbox

Radio buttons


● Auswahl 1
○ Auswahl 2

By clicking on the „Submit Button" the form contents are sent from the browser to the web server.


Senden

Web pages that are constructed in pure HTML have a decisive disadvantage - once loaded in the browser, there is no updating without user intervention.

Therefore, web pages today are built dynamically and no longer in pure HTML.

Dynamic websites

The requirements for web pages have changed in recent years in such a way that HTML as a formatting markup language is no longer sufficient to meet these requirements.

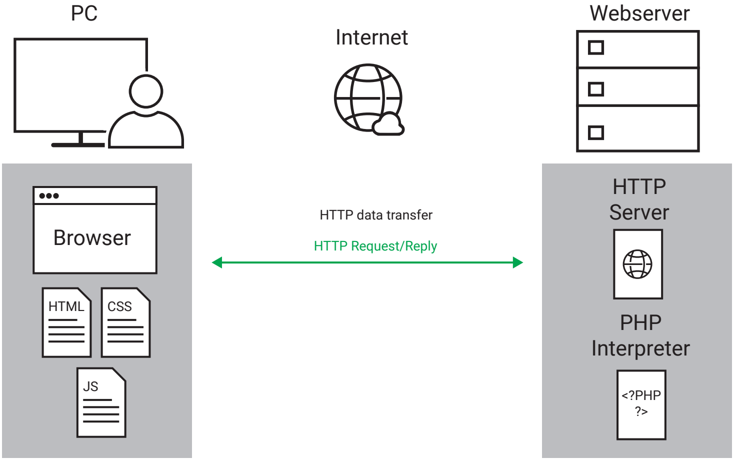
Connections to everything from social media to online shopping and technical applications like Smart Home and IoT require a constant data exchange between browser and server and a permanent update of the display.

In order to achieve this, modern web design uses various techniques, some of which are interlinked.

Even if the part in the browser visible to the user is a web page, one must rather speak of a web application.

Web applications with HTML, CSS, JavaScript and PHP

Today's web applications always have a browser-side and a server-side part. A common combination is HTML, CSS and JavaScript on the browser side and PHP on the server side.



HTML

In the example shown here, the HTML file actually only lists the elements to be displayed in HTML format. No properties such as position or color are defined for the individual elements, as is the case with classic HTML. Instead, the elements are given an ID (identification description) or assigned to a specific display class.

```
<span id="footer1" class="blueline">Fußzeile</span>
```

There is also a reference to the CSS file to be used and the JavaScript to be used.

```
<head>
  <title>Dynamische  Webseite</title>
  <script language="javascript" type="text/javascript" src="jscript.js">
  </script>
  <link rel="stylesheet" href="style.css"/>
</head>
```

CSS - Cascading Stylesheet

In the CSS file the visual properties of the elements listed in the HTML file are assigned.

This can be done individually for individual elements,

```
#footer1 {
  width:100%;
  height: 20px;
}
```

But you can also define assignments for groups of elements in the form of display classes.

```
.blueline {
  color: blue;
}
```

Separating the style properties from the actual HTML file is especially useful for larger Web applications, since the style definitions can be applied to more than one Web page.

CSS files have the name extension .css.

JavaScript

JavaScript is a programming language that is carried out in the browser. The JavaScript code can be integrated in the HTML file or can be stored in a separate JavaScript file.

The difference between a markup language (like HTML) and a programming language is essentially that a programming language takes case differences into account.

The code of a markup language is executed rigidly from top to bottom. In a programming language when, how and what is carried out is decided according to predefined conditions.

In the case of JavaScript, the appearance and properties of elements that are already displayed in the browser can subsequently be changed. JavaScript can also be used to exchange additional data with the web server.

JavaScript files have the name extension .js.

AJAX - Asynchronous JavaScript and XML

AJAX describes the technique of using JavaScript to communicate with the web server from a web page that has already been loaded using HTTP requests. The XML in the name AJAX comes from the fact that many web applications exchange data with the server in XML format, but this is not a must.

PHP

Like JavaScript, PHP is a script language. However, the PHP program code is already executed on the server. The .php extension tells the server that the file called up by the browser contains PHP parts. First of all the server processes the PHP code, which ultimately determines which contents are transferred to the browser.

A very simple PHP file could look like this:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Wiesemann &amp; Theis GmbH</title>
  </head>
  <body>
    <?php
      echo "Willkommen bei WuT.de";
    ?>
  </body>
</html>
```

PHP files can be structured like a normal website written in HTML. The PHP parts are enclosed in corresponding tags.

- PHP Start `<?php`
- PHP End `?>`

The PHP part marked in this way is not sent to the browser. The browser would receive the following page source code for the above example. :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Wiesemann &amp; Theis GmbH</title>
  </head>
  <body>
    Willkommen bei WuT.de
  </body>
</html>
```

By executing the PHP code, simply “Welcome to WuT.de” is displayed.

PHP files do not necessarily have to contain HTML parts. Nevertheless the PHP source code is always enclosed by the PHP tags.

In the given example the use of PHP does not make much sense yet. But PHP can do much more. A very important feature of PHP is the database support. PHP can directly access databases such as MySQL. As a result it offers ideal conditions for the realization of online shops.

But PHP can also be used to establish and manage normal TCP or UDP socket connections. In the industrial sector, data from devices that do not have a web interface can also be displayed in the browser.

Besides the possibility of creating dynamic web applications, as is shown here as an example, there are other ways, which we would like to introduce briefly in the following.

Server side programs

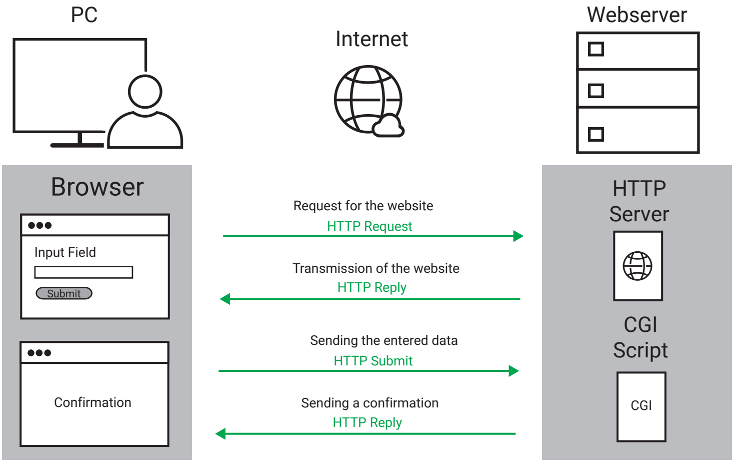
CGI - Common Gateway Interface

For a long time, the use of CGI scripts was the most commonly used method for displaying interactive content in the browser for triggering actions.

Via CGI, programs can be executed on the web server from the browser.

A hyperlink, a submit button, or direct entry of the URL is used to call the corresponding program and, if necessary, the required parameters are transferred.

A classic example are HTML forms that are filled out by the user. If the user clicks the Submit button, the input is transferred to the Web server via HTTP using the POST command. The specified CGI script is started and processes the input further.



Other possible applications are visitor counters, guest books, discussion forums, database access or search engines.

CGI scripts can be created in all common programming languages. It is important that the web server supports the chosen language.

In practice, the programming language Perl has become generally accepted for the creation of CGI scripts.

PHP

PHP has now replaced CGI as the most widely used method for displaying interactive content. A detailed description of PHP was already given in the previous section.

ASP - Active Server Pages

ASP is a technique created by Microsoft to dynamically display web pages. ASP-based web pages consist, as does PHP technology, of classic HTML components and scripts that are executed on the server side when the web page is called up. An interpreter on the WWW server is controlled by these scripts and generates web pages in HTML format.

VBScript (Visual Basic Syntax) or JScript (Java Syntax) are usually used as script languages. An advantage of this technique is that DLLs and AktivX components installed on the server can be used. Dynamic Link Libraries and AktiveX components are ready-made, outsourced program functions that relieve the programmer of work, since corresponding, often complex functionalities do not have to be programmed by the user.

The disadvantage of ASP lies in the server operating system requirements. Originally, ASP support was only available on Microsoft server systems. For some time now, however, third-party manufacturers have been providing ASP variants for Linux servers.

ASP-based websites can be recognized by the extension ".asp" in the file name.

The classic ASP has now been replaced by Microsoft with ASP.NET.

Browser side programs

JavaScript

We have already described JavaScript in the previous section.

Here is a short example for more details:

Public web presences are represented by domain names. One and the same website can be accessible via several domain names, for example via a "de" domain and a "com" domain. The following code evaluates whether a website was accessed via the "com" domain or the "de" domain and displays itself in English or German.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>urltest</title>
  </head>
  <body>
    <script language="JavaScript">
      if (location.hostname == "www.web-io.com")
        document.write("welcome at WuT");
      else
        document.write("Willkommen bei WuT");
    </script>
  </body>
</html>
```

AJAX - Asynchronous JavaScript and XML

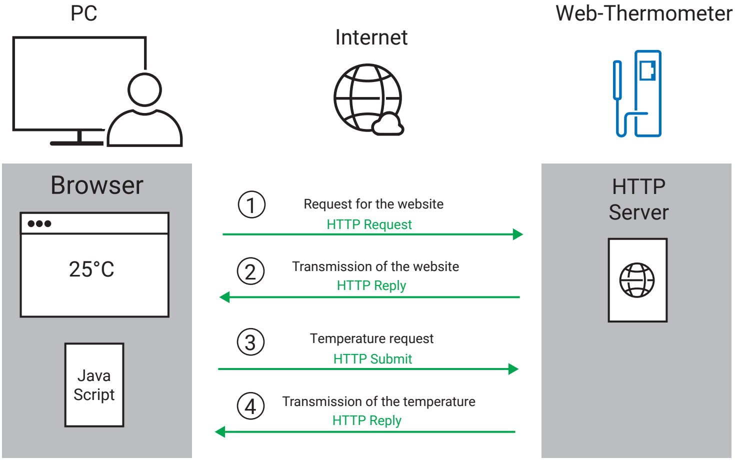
AJAX has also already been briefly introduced. Since AJAX is currently the most popular technique for exchanging data with the web server from a website that has already been loaded, some additional information will be given here.

The core of AJAX is the HTTP request method offered in current versions of JavaScript. Using this method JavaScript can request data from the web server even after loading and displaying a web page. If the server supports it, data can be requested in XML format. Alternatively, the transfer of text formats is also supported.

JavaScript handles both the later retrieval of data and the updating of the browser display.

JavaScript cannot maintain a permanent connection to the web server. However, a self-updating process visualization in the browser can be realized by cyclical reloading of process data.

As a short example, here is a display of the temperature measured by a W&T web thermometer:



1. the web page stored by the user on the web thermometer is entered and called up in the browser as URL. The browser sends a corresponding HTTP request.
2. The web thermometer sends the web page including the JavaScript component to the browser.
3. The JavaScript is executed and requests the current temperature from the web thermometer via HTTP request.
4. The web thermometer sends the current temperature to the browser, where the JavaScript evaluates the received data and updates the display accordingly.

Steps 3. and 4. are repeated cyclically as long as the web page remains open in the browser.

For security reasons, this technique only allows data to be reloaded only from the server from which the original web page was loaded with JavaScript. This is to prevent a user's browser from being used to gain unrecognized access to the web presence of third parties.

With newer web servers, it is possible to allow such cross origin requests from specific addresses.

Java-Applets

Just a few years ago, Java applets were the tool of choice when it came to creating dynamic web pages. Today Java applets have been almost completely replaced by AJAX technology.

Java applets are compiled programs and require certain plug-ins in the browser, i.e. additional functions that are not inherently available in current browsers.

Compiled programs are programs where the files come as binary data, so unlike JavaScript, for example, they are not readable.

Binary files and plugins are a security risk, because you cannot see which functions are actually integrated.

In many company networks Java applets are therefore blocked by browsers and firewalls.

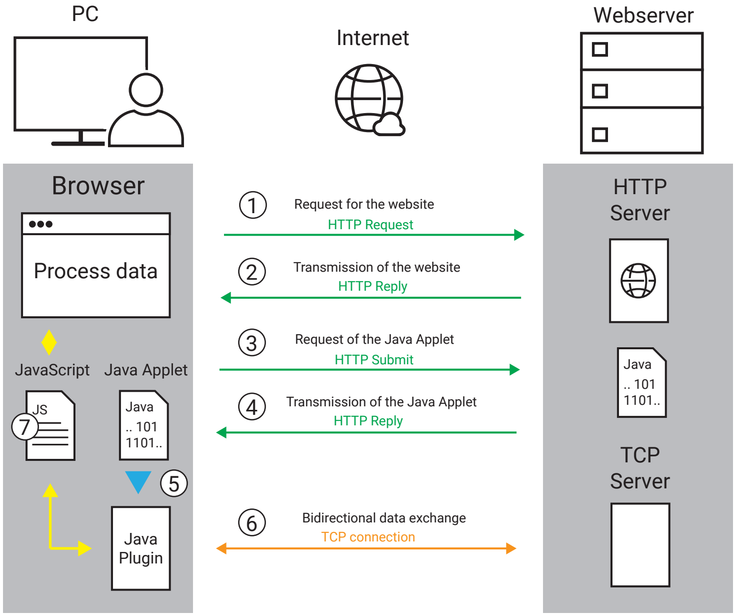
Nevertheless, there are cases in which Java applets can be used sensibly if they come from a secure source.

One disadvantage of AJAX technology, on the other hand, is that there is no permanent connection to the server. The server cannot send any information to the browser on its own, communication can only take place via HTTP.

But with a Java applet this would be possible. The link to the Java applet to be used is included in the <body> section of the website: The corresponding `<applet>` tags are used for this. Additionally, parameters for the applet can be defined.

```
<applet archive="A.jar" code="A.class">
  <param name="getalldata" value="1">
</applet>
```

Here is an example where process data is to be exchanged with the browser via a normal TCP connection



1. The URL of the web page is entered and accessed in the browser. The browser sends a corresponding HTTP request.
2. The server sends the web page including a JavaScript part to the Browser.
3. The browser finds the reference to the Java applet and sends a second HTTP request to the server.
4. The server sends the Java applet to the browser.
5. The Java applet is loaded into the Java engine and started there. The Java engine works as a plugin in the browser.
6. The Java engine now establishes the TCP connection to the server.
7. The JavaScript embedded in the web page exchanges send and receive data of the TCP connection with the Java engine if required. The JavaScript evaluates the received data and displays it in the browser.

In addition to the possibility of using TCP and UDP as communication paths, Java applets can also contain visual elements (displays, diagrams, characteristic curves, ...) that are to be displayed in a web page.

Even this simple example shows that the development and integration of websites that use a Java applet are very complex. As mentioned previously there are also security risks involved. So, whether Java applets are the right way to go for the desired application should be considered on a case-by-case basis.

Responsive web design

The demands on websites and the technology of web design have changed a lot over the last years.

In the early days of the internet, websites created in HTML were rather bulky and were mostly designed for low resolution PC monitors.

With JavaScript a certain dynamic was added and the contents became more intricate in their presentation, but were still designed for viewing on PC monitors.

Today, most users use smartphones and tablets in addition to the PC to call up websites. Size and resolution of the displays vary considerably. This results in completely new challenges for web design.

The browsers used on smartphones automatically scale the web pages to the size of the display. But this often makes the content so small that it is no longer readable. If you zoom the display larger, you can no longer see the entire content and the website becomes unclear.

Different websites for different display sizes

Initially, the web designers responded by creating different web pages with the same content for different display devices. Using Javascript, they determined which end device would call up the web page and then send the appropriate web page to the browser.

However, this makes the maintenance of web pages very time-consuming, as two to three web pages have to be adapted each time the content changes.

Responsive websites

A much more elegant method to meet the different display sizes is reponsive web design.

As already described, the content and design of a website can be separated by cascading stylesheet files. As with normal websites, an HTML or PHP file is created in which the content is defined.

In addition, there is an entry in the header of the web page which defines that the entire display width is used with a 1:1 scaling.

```
<head>
    ........
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    ........
</head>
```

There is a CSS file to match. Current browsers accept areas within the CSS file with style presets for different display sizes.

The syntax always starts with @media followed by other parameters. Here is an example for a display width between 480 pixels and 632 pixels

```
@media only screen and (min-width: 480px) and (max-width: 632px) {
  .........
}
```

The corresponding style information then follows, enclosed in curly brackets.

In this way, the size and position of display elements, font size and much more can be individually specified for the corresponding display size.

The display in the browser automatically adapts to the screen or the size of the browser window.

A further advantage of this technology is that with

```
@media only print ....
```

you can define how the printout of a web page should look like.

There are also numerous tutorials on the internet on the subject of Responsive Web Design.

Network ABC

10Base2 – 10Mbit/s BASEband 200 (185)m/segment

In the 80s and 90s, 10Base2 was used as an Ethernet topology on a coaxial basis for networking PCs and other network components.

The maximum transmission rate was 10Mbit/s.

Other names for 10Base2 were Cheapernet or Thin-Ethernet.

Coax cables with 50 Ohm impedance in a thin and flexible design were used to connect the individual stations in bus topology. The beginning and end of a segment had to be terminated with 50 Ohm termination resistors.

The weakness of the physical bus topologies of Ethernet was that an interruption of the cable - e.g. by pulling a connector - caused the entire network segment to come to a standstill.

10Base5 – 10Mbit/s BASEband 500m/segment

10Base5 is the original Ethernet specification. The cabling is based on a coaxial bus cable with 50 Ohm impedance and a maximum permissible length of 500m (Yellow Cable).

The network participants were connected via external transceivers, which used vampire clamps to tap the signals directly from the bus cable without interrupting it. The terminal devices were connected to the transceiver via an additional cable.

The use of a relatively high-quality cable without any interruptions by plug connectors results in the advantages of a large segment length and a high number of possible connections per segment (max. 100).

The thickness and inflexibility of the Yellow Cable and the additional costs caused by external transceivers are the main disadvantages of 10Base5 and have probably contributed significantly to the introduction of 10Base2.

10BaseT – 10Mbit/s BASEband Twisted Pair

With the definition of 10BaseT, the physical topology is separated from the logical one. The cabling is star-shaped, starting from a hub as the central active component. An at least two-pair Category 3 cable with 100 Ohm impedance is used, in which the data is transmitted separately in the transmit and receive direction. 8-pole RJ45 connectors are used, in which the pairs are connected to pins 1/2 and 3/6. The maximum length of a segment (= connection from the hub to the terminal device) is limited to 100m. The 10BaseT topology has its origins in the USA. It made it possible to make use of the telephone wiring commonly used there for network operation. For Germany this advantage was not applicable, because star 4 cables were laid here for telephony and this did not meet the requirements of category 3.

All physical bus structures face a standstill of the entire segment when cables are interrupted or plugs are disconnected. When it come to 10BaseT only one workstation is affected. (*see* 📄 *13)*

100BaseFX – 100Mbit/s BASEband Fiber Exchange

This is the Ethernet standard for star-shaped fiber optic cabling with a transmission rate of 100Mbit/s. To transmit the Ethernet signal, light pulses with a wavelength of 1300nm are fed into a 50µm or 62.5µm multimode fiber. The maximum segment length is 2km.

100BaseT4 – 100Mbit/s BASEband Twisted 4 Pairs

100BaseT4 specifies an Ethernet transmission with 100Mbit/s. Like 10BaseT. It is a physical star structure with a hub as the center. It also use a category 3 cable with 100 Ohm impedance, RJ45 connectors and a maximum length of 100m. The tenfold transmission speed of 100Mbit/s with simultaneous adherence to the category 3 bandwidth of 25MHz is also achieved by using all four wire pairs. With 100BaseT4, 3 pairs are always used simultaneously for each data direction.

100BaseTX – 100Mbit/s BASEband Twisted 2 Pairs

100BaseTX specifies 100Mbit/s transmission on 2 wire pairs via cabling implemented with category 5 components. Cables, RJ45 wall outlets, patch panels etc. must be designed for a transmission frequency of at least 100MHz according to this category.

1000BaseT

1000BaseT is also known as Gigabit Ethernet. 1000Mbits/s can be transmitted over a maximum distance of 100m via cables and components that correspond to at least category 5. (*see* 🗎 *14)*

1000BaseBX, 1000BaseLX, 1000BaseSX, 1000Base ZX/EZX

In addition to the wired 1000BaseT, there are various Gigabit Ethernet standards that use fiber optics as a transmission medium.

10GBaseT

With transfer rates up to 10Gbits/s, 10GBaseT is used as a backbone, i.e. background cabling between switches. Cables of at least category 6 are required. (*see* 🗎 *15)*

10GBaseER, 10GBaseEW, 10GBaseLX4, 10GBase LW, 10GBaseSW

Transmission standards for 10GBase over optical fiber.

ABAP - Advanced Business Application Programming

ABAP is a programming language developed by SAP to program the SAP software environment individually.

Administrator

System administrator who has unlimited access rights in the local network and is responsible for the administration and maintenance of the network. Among other things, the administrator assigns the IP addresses in his network and must ensure that each IP address is unique.

ADSL

Asynchronous DSL connection with different speeds for upload and download. (*see* 🗎 *201)*

AES

Advanced Encryption Standard is a symmetrical encryption algorithm that is used, for example, when transmitting web pages via HTTPS.
(*see* 🗎 *154)*

AJAX - Asynchronous JavaScript and XML

AJAX is a JavaScript programming technique that allows the content of a web page to be updated dynamically by reloading data. (*see* 🗎 *213ff)*

ARP – Address Resolution Protocol

Via ARP the Ethernet address belonging to an IP address of a network participant is determined. The determined assignments are managed on each individual computer in the ARP table. In Windows operating systems, the ARP table can be influenced by means of the ARP command. Properties and parameters of the ARP command in the DOS box:

- ARP -A lists the entries of the ARP table
- ARP -S <IP address> <Ethernet address> adds a static entry to the ARP table
- ARP -D <IP address> deletes an entry from the ARP table

ARP is defined in the internet standard RFC-826. *(see 🗎 26)*

ASCII-Coding

As early as 1963, the American Standard Code for Information Interchange defined which character is encoded with which 7-bit value during data transmission.

0x0X 0000	0x1X 0001	0x2X 0010	0x3X 0011	0x4X 0100	0x5X 0101	0x6X 0110	0x7X 0111	hex. dual
NUL 0	*DEL* 16	Space 32	0 48	@ 64	P 80	` 96	p 112	0xX0 0000
SOH 1	*DC1* 17	! 33	1 49	A 65	Q 81	a 97	q 113	0xX1 0001
STX 2	*DC2* 18	" 34	2 50	B 66	R 82	b 98	r 114	0xX2 0010
ETX 3	*DC3* 19	# 35	3 51	C 67	S 83	c 99	s 115	0xX3 0011
EOT 4	*DC4* 20	$ 36	4 52	D 68	T 84	d 100	t 116	0xX4 0100
ENQ 5	*NAK* 21	% 37	5 53	E 69	U 85	e 101	u 117	0xX5 0101
ACK 6	*SYN* 22	& 38	6 54	F 70	V 86	f 102	v 118	0xX6 0110
BEL 7	*ETB* 23	' 39	7 55	G 71	W 87	g 103	w 119	0xX7 0111
BS 8	*CAN* 24	(40	8 56	H 72	X 88	h 104	x 120	0xX8 1000
TAB 9	*EM* 25	) 41	9 57	I 73	Y 89	i 105	y 121	0xX9 1001
LF 10	*SUB* 26	* 42	: 58	J 74	Z 90	j 106	z 122	0xXA 1010
VT 11	*ESC* 27	+ 43	; 59	K 75	[91	k 107	{ 123	0xXB 1011
FF 12	*FS* 28	, 44	< 60	L 76	\ 92	l 108	| 124	0xXC 1100
CR 13	*GS* 29	- 45	= 61	M 77	] 93	m 109	} 125	0xXD 1101
SO 14	*RS* 30	. 46	> 62	N 78	^ 94	n 110	~ 126	0xXE 1110
SI 15	*US* 31	/ 47	? 63	O 79	_ 95	o 111	*DEL* 127	0xXF 1111

In addition to the printable characters, the coding for the values 0 to 32 and the value 127 contains functional respectively control codes:

0 NUL NULL character - is used to identify sections in data streams
1 SOH Start Of Heading - the first character of the message header
2 STX Start Of Text - the first character of a message
3 ETX End Of Text
4 EOT End Of Transmission
5 ENQ Enquiry - request for a response
6 ACK Acknowledgment - data reception acknowledgment
7 BEL Bell

8 BS Backspace
9 TAB Horizontal tab
10 LF Line Feed
11 VT vertical tab
12 FF Form Feed - Form Feed
13 CR Carriage Return - moves the cursor to the first position in the line
14 SO Shift out - finishes special handling that was started with Shift
15 SI Shift in - the following characters will be treated specially
16 DLE Data Link Escape
17 DC1 Device Control 1 - calls a predefined device function
18 DC2 Device Control 2 - calls a predefined device function
19 DC3 Device Control 3 - calls a predefined device function
20 DC4 Device Control 4 - calls a predefined device function
21 NAK Negative Acknowledgment - data reception not correct
22 SYN Synchronous Idle
23 ETB End Of Transmission Block - end of a transmission block
24 CAN Cancel
25 EM End Of Medium - No further data processing possible
26 SUB Substitutes
27 ESC Escape - the following characters have a special meaning
28 FS File Separator
29 GS Group separator
30 RS Record separator
31 US Unit separator
127 DEL Delete - dates from the time when data was stored on punch cards. The seven bits were coded with 1 = hole and 0 = not hole. With DEL all 7 holes were punched. So every character could be overwritten and thus marked as invalid

ASN.1

Format for building SNMP MIB files. *(see 📄 89)*

AUI – Attachment Unit Interface

Interface for connecting an external Ethernet transceiver.

Separated according to transmit, receive and collision information, the data is provided by the transceiver on a 15-pin D-SUB connector. The terminal device is connected via an 8-core TP cable with a maximum length of 50 m. Whereas in the past the AUI interface was mainly used to connect terminal devices to 10Base5 transceivers (yellow cable), it is now more commonly used to connect to fiber optic transceivers or similar.

Backbone

Backbone is the background cabling between sites or switches. Often a faster transmission method is chosen for the backbone connections than for the connections within the local network.

Base64

An encoding method to transmit binary data such as images with 7 bit technology. *(see 🗎 120)*

Bit

The bit is the smallest memory unit in computer technology and can assume one of the two states 1 or 0. *(see 🗎 9)*

Bluetooth

Radio standard to connect terminals with each other over short distances.

Binary data

Binary data is data where each byte may have values between 0 and 255. *(see 🗎 116)*

BNC – Bayonet Neill Concelmann

The BNC connector is a bayonet lock for connecting two coaxial cables. BNC connectors are used in 10Base2 networks to mechanically connect RG-58 cables (Cheapernet).

BootP – Boot Protocol

This older protocol for booting PCs without a hard disk over the network is the predecessor of DHCP. Even modern DHCP servers still support BootP requests. Today, BootP is primarily used to assign an IP address to embedded systems. For this purpose, a reserved entry must be stored on the DHCP server, in which a fixed IP address is assigned to the MAC address of the embedded system.

Bridge

Bridges connect sub-networks with each other and decide which packets are allowed to pass the bridge and which are not based on the Ethernet address. The bridge takes the necessary information from tables, which, depending on the model, must be entered by the administrator or are dynamically created by the bridge itself. *(see Router)*

Broadcast

Broadcast is a broadcast call to all network participants. A typical broadcast application is the ARP request (see ARP). Other protocols - such as RIP or DHCP - also use broadcast messages.

Broadcast messages are not forwarded via routers or bridges.

Broker

A broker is a server within an MQTT application that transmits data via publish/subscribe procedures. *(see 🗎 133)*

Browser

Client program with a graphical user interface that allows the user to display web pages and use other services on the internet.
(see 🗎 206)

Bus system

In a bus system, several terminal devices share a single data line (bus line). Since only one terminal device may use the data line at any given time, bus systems always require a protocol to control access rights. Classical bus systems are the Ethernet topologies 10Base2 and 10Base5.

Byte

A byte consists of 8 bits and is the smallest amount of data that computers can process. The width of 8 bits means that one byte can store or transmit numerical values between 0 and 255. There is more of this in the chapter number systems. *(see 🗎 9)*

Cache

A cache is a buffer, such as the one used in a browser to temporarily hold web pages and other content. If a particular content is retrieved from the web server several times at short intervals, the browser does not request the required data from the server again, but takes the already loaded data from the cache. *(see 🗎 100)*

Cheapernet

Another name for Ethernet based on 10Base2.

Checksum

A checksum can be formed over the content of transmitted or stored data according to a predefined algorithm. Before the data is processed further, the same algorithm can be applied again to check whether the content is unchanged. *(see 🗎 149)*

Cipher Suites

A cipher Suite is a combination of methods for authentication, checking integrity and encryption of data. (*see* 🗎 *173)*

Client

Computers or applications that use the services of so-called servers. Server services can be, for example, the provision of a COM or printer interface in the network, but also Telnet and FTP (*see* 🗎 *28)*

Client/server architecture

System in which the client connects to a server to take advantage of services offered by the server. Some server applications can serve multiple clients simultaneously. (*see* 🗎 *28)*

Com-Server

Terminal device in TCP/IP Ethernet networks that provides interfaces for serial devices via the network. *(see. https://www.wut.de/e-58665-ww-daus-000.php)*

Community String

The community string is a kind of password that is sent with every SNMP query. (*see* 🗎 *91)*

Cookies

User information, e.g. customer number or similar, which the browser caches in such a way that it is still retained after the next start when visiting the same website.

Cross Link cable

Network cable in which the cable wires are crossed for sending and receiving Using a cross-link cable, network terminals (which do not support autocrossing) can be connected directly without an additional switch.

DHCP – Dynamic Host Configuration Protocol

Dynamic allocation of IP addresses from an address pool.

DHCP is used to configure PCs in a TCP/IP network automatically - i.e. without manual intervention - centrally and thus uniformly. The system administrator determines how the IP addresses are to be assigned and specifies the time period over which they are assigned. DHCP is defined in the internet standards RFC 2131 (03/97) and RFC 2241 (11/97). (*see* 🗎 *59)*

DDNS – Dynamic Domain Name Service

DNS service that also supports name resolution for those network participants that obtain their IP address dynamically via DHCP. *(see* 🗎 *68)*

DNS – Domain Name Service

Network participants are addressed on the internet via numerical IP addresses. However because names are easier to remember than numbers, the DNS was introduced.

DNS is based on a hierarchical system: each name address is identified by a top-level domain (“de”, “com”, “net”, etc.) and within this domain by a sub-level domain. Each sub-level domain can (but does not have to) contain subordinate domains. The individual parts of this name hierarchy are separated by dots.

If the user specifies a domain name for addressing, the TCP/IP stack requests the corresponding IP address from the next DNS server.

It is sensible to give network resources a domain name that provides some context for the service offered or the company name of the provider. For example, wut.de can be resolved into the top-level domain de (= Germany) and the sub-level domain wut (= Wiesemann & Theis GmbH) *(see* 🗎 *64)*

DNS Server

DNS servers provide the service of resolving a domain name into an IP address on the internet.

DOS Disk-Operation-System

Early operating system from Microsoft on command line basis.

Driver

Software to integrate / embed hardware components or peripheral devices into an operating system

DynDNS

In case most internet accesses, the connected terminal device receives an IP address from the address pool of the internet provider at the time of dial-up. Since this temporary IP address is not known to the outside world, such mobile devices are normally not addressable from the internet. A name can be assigned to such an internet user via DynDNS. DynDNS updates the assignment between name and temporary IP address as soon as the participant goes online, so that accessibility via the name is possible. *(see* 🗎 *69)*

EDGE - Enhanced Data Rates for GSM Evolution

EDGE is a further development of GSM technology and is based on more efficient data modulation and compression methods. GPRS becomes E-GPRS (Enhanced GPRS) with EDGE and allows data rates up to 220Kbit/s (download) and 110Kbit/s (upload).

Since EDGE is only an extension of GSM, terminals of both technologies can be operated in the same network.

E-Mail

Electronic mail via internet and Intranet. *(see 🗎 103)*

E-Mail address

An e-mail address is required to send electronic mail to a user and is always composed of the user's mailbox name and the target domain, separated by the @ sign. An example: info@wut.de designates the info mailbox on the mail server of W&T. *(see 🗎 104)*

Embedded System

An embedded system is a microprocessor-controlled module which, as an embedded part of a device or machine, processes data in the background and controls processes if necessary.

ERP System

Enterprise Resource Planning System - this is a software solution that helps companies to use capital, operating resources and personnel as efficiently as possible. The best-known provider in this area is SAP.

Ethernet

Ethernet is currently the most commonly used technology for local networks. *(see 🗎 12ff)*

Ethernet address

The unchangeable physical address of a network component in the Ethernet.

(see 🗎 22)

Fast-Ethernet

Fast Ethernet is virtually an upgrade of the 10BaseT topology from 10Mbits/s to 100 Mbit/s. (see.100BaseT4 and 100BaseTX)

Fiber optic

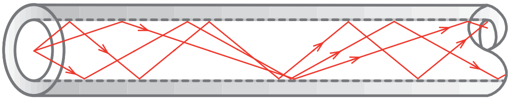
In network and communications engineering, fiber optic - or FO for short - are increasingly used as a communication medium. Especially in network technology, FOs can be used to bridge much greater distances than conventional copper cabling. In addition, data transmission via FO is resistant to electrical influences such as lightning and coupling of external and interfering signals.

Electrical signals are converted into light signals and fed into the fiber optic cable via fiber optic transmitters. Glass fibers are usually used as the transmission medium.

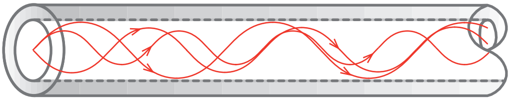
A distinction is made between multi mode fibers, mono mode fibers and plastic optical fibers.

Multimode fiber

Multimode fibers have a fiber diameter of 62.5 µm or 50 µm. Since light propagates in all directions if possible, it takes different signal paths within the fiber (therefore multi mode). Due to the different angles of reflection, the light travels shorter and longer paths until it reaches the receiver.



Such multimode optical fibers are also known as step index fibers. In addition to the step index fibers there are gradient index fibers. In these fibers, the light also propagates in different directions. However, due to a special optical property, the light rays are gently deflected and not reflected from the edge at an angle as in the case of the step index fiber.

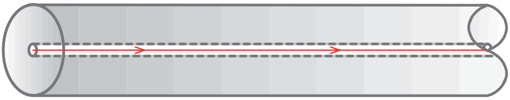


Gradient index fibers have a higher bandwidth than step index fibers and therefore allow higher signal speeds.

Depending on the signal to be transmitted, both multimode fiber types can bridge distances of up to several kilometers (e.g. max. 2km for 100BaseFX).

Monomode fiber

Mono mode fibers - often also called single mode fibers - have a fiber diameter of 3-9µm. Due to the small fiber diameter the light can only propagate on one signal path (therefore single mode).



Single mode fibers allow distances of up to 50km depending on the signal to be transmitted.

Due to the small fiber diameter of max. 9µm (a human hair has a diameter of approx. 100µm) the processing of single mode fibers is much more complex than is the case with multimode fibers.

Plastic optical fibers

Plastic optical fibers are rarely used in network technology. In order to complete the picture, here are a few words about this technology.

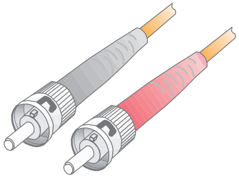
Polymer fibers are usually used for data transmission via plastic optical fibers. With a common diameter of 1mm polymer fiber is definitely a multimode fiber. The high attenuation of the polymer material limits the maximum length of the transmission distances to 20 - 100m. The main application for polymer fiber optics is therefore the transmission of serial signals such as RS232 or RS422/48.

Connector types

There is another variance in the FO plug connections. Here there are three basic procedures:

Connectors with bayonet locking, connectors with union nut and push-pull connectors with spring lock

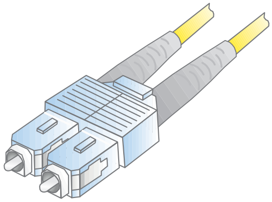
.



ST plug

Fiber optic type:	Multimode, Monomode
Locking:	Bayonet lock
Application:	LAN, WAN, serial signals

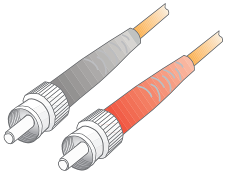
For a long time, the ST connector was the most used in the network and industrial sector. Twist protection and bayonet lock make the ST connector safe and easy to use.



SC plug

Fiber optic type:	Multimode, Monomode
Locking:	Push-Pull
Application:	LAN, WAN, serial signals

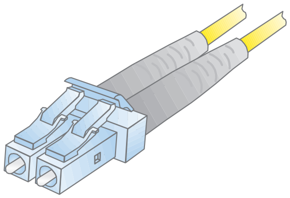
Due to its simple push/pull handling and duplex capability, the SC connector has now replaced ST technology as the most common.



SMA plug

Fiber optic type:	Multimode, Monomode
Locking:	Union nut
Application:	LAN

The SMA connector was used in the early days of fiber optic technology. The lack of twist protection and the danger of over-tightening often led to damage to the fiber, which is why SMA technology is of little importance today.



LC plug

Fiber optic type:	Multimode, Monomode
Locking:	Push/Pull
Application:	LAN, WAN

Due to its compact design, the LC connector is mainly used for connection to switches and other active network components.

At this point we have only presented the four most commonly used connector systems. A complete list of all fiber optic connector variants would go beyond the scope of this book.

Fiber standards

In addition to the different connectors, there are various standards for Ethernet via fiber optics, which differ in particular with regard to data transmission rates.

Here is a brief overview:

	Data rate	FO-Type	Wavelength	Fiber	Fibers	Max. length	Plug	IEEE-
10BaseFB	10 Mbit/s	MM	850nm	62,5/125µm	2	2 km	ST	802.3j
10BaseFL	10 Mbit/s	MM	850nm	62,5/125µm	2	2 km	ST	802.3j
100BaseFX	100 Mbit/s	MM	1310nm	62,5/125µm	2	2 km	ST, SC	802.3u
100BaseSX	100 Mbit/s	MM	850nm	50/125 µm, 62,5/125µm	2	300 m	ST, SC, LC	TIA-785
1000BaseSX	1 Gbit/s	MM	850nm	50/125µm	2	550 m - 1 km	ST, LC, SC	802.3z
1000BaseLX	1 Gbit/s	MM SM	1310nm	50/125µm, 9/125µm	2	550 m- - 5 km	SC, LC	802.3z
1000BaseLX-10	1 Gbit/s	MM SM	1310nm	50/125µm, 9/125µm	2	550 m- -10 km	LC	802.3ah
1000BaseEX	1 Gbit/s	SM	1310nm	9/125µm	2	40 km	SC, LC	-
1000BaseZX/-EZX	1 Gbit/s	SM	1550nm	9/125µm	2	70 km	SC, LC	-
1000BaseBX-10	1 Gbit/s	SM	1310nm/1550nm, 1490nm/1550nm	9/125µm	1	10 km	LC	802.3ah

	Data rate	FO-Type	Wave-length	Fiber	Fibers	Max. length	Plug	IEEE-
10GBase-R	10 Gbit/s	MM	850nm	50/125µm, 62,5/125µm	2	33m - 400m	SC, LC	802.3ae
10GBaseSW	10 Gbit/s	MM	850nm	50/125um, 62,5/125um	2	33 m - 400 m	SC, LC	802.3ae
10GBaseLX4	10 Gbit/s	MM SM	1269.0nm/ 1282.4nm, 1293.5nm/ 1306.9nm, 1318.0nm/ 1331.4nm, 1342.5nm/ 1355.9nm	50/125µm, 9/125µm	1	300 m -10 km	SC	802.3ae
10GBaseLR	10 Gbit/s	SM	1310nm	9/125µm	2	10 km	SC, LC	802.3ae
10GBaseLW	10 Gbit/s	SM	1310nm	9/125µm	2	10 km	SC, LC	802.3ae
10GBaseER	10 Gbit/s	SM	1550nm	9/125µm	2	40 km	SC, LC	802.3ae
10GBaseEW	10 Gbit/s	SM	1550nm	9/125µm	2	40 km	SC, LC	802.3ae
10GBase-PR	10 Gbit/s	SM	1270nm/ 1577nm	9/125µm	1	20 km	SC	802.3av

Fieldbus

Bus system for industrial use (see bus systems)

Firewall

A firewall is a network component that, similar to a router, connects two networks with each other. According to predefined rules the firewall filters which contents maybe transfered from one network to the other. Firewalls are often part of routers that connect an internal network (intranet) to a public network (e.g. the internet). *(see 🗎 47)*

FTP – File Transfer Protocol

FTP is a protocol based on TCP/IP, which makes it possible to transfer entire files between two network participants. *(see 🗎 79)*

Gateway

Gateways - like bridges and routers - connect different networks with each other. While bridges and routers may implement the physical type of network (e.g. Ethernet/ISDN), but leave the actual protocol (e.g. TCP/IP) untouched, gateways offer the possibility of creating access to networks that do not conform to the protocol (e.g. TCP/IP on Profibus). Among other things, a gateway has the task of translating different communication protocols.

Be Careful!: During network configuration in Windows operating systems, the input of a gateway is also required. However, this specification refers to a router that may be present in the network! (*see* 🗎 *39)*

GPRS - General Packet Radio Service
GPRS is a standard for data transmission based on GSM.

Although GSM focuses on the transmission of voice data, GSM offers the possibility to transmit data via GPRS. There are also 8 channels available for data transmission per frequency. In telephony, a connection is established for the duration of a call, blocking one channel in each direction.
In the case of data transmission with GPRS, a channel is only blocked if data is actually being sent and can therefore be used by several participants with a time delay. The parallel use of several channels by one subscriber is also permitted. Thus, transmission rates of up to 54Kbits/s can be achieved, which is roughly equivalent to the transmission speed of analog modems.
(see 🗎 *202ff)*

GSM - Global System for Mobile Communications
GSM is the original and first standard for digital mobile telephony and the technical basis of the D and E networks. GSM operates in a frequency range between 890MHz and 960MHz. Within this range there are 124 channels per transmission direction. Within each individual frequency 8 channels share the transmission time. By means of data compression procedures, 8 subscribers can thus telephone simultaneously over the same transmission or reception frequency.
(see 🗎 *202ff)*

HSPA, HSDPA/HSUPA - High Speed Packet Access
HSPA is a further development of UMTS. It uses the same frequency band and the same transmission technology on the provider side. However, a significantly improved modulation and coding method is used. In order to achieve maximum efficiency, the download (HSDPA) process is different from the upload (HSUPA) process. The reason for this is that the transmitting equipment on the provider side has significantly more transmitting power available than the customer's mobile device.

When downloading with HSDPA, up to 42Mbit/s can be reached. Up to 5.8Mbit/s are possible when uploading. (*see* 🗎 *202ff)*

HTML – Hypertext Markup Language
Markup language that uses keywords to specify how content is displayed in the browser, where multimedia elements can be found and which elements are linked and how. *(see* 🗎 *208ff)*

HTTP – Hypertext Transfer Protocol

The HTTP protocol is based on TCP and regulates the request and transfer of web content between HTTP server and browser. (*see* 🗎 *94)*

Hyperlink

Reference to other websites or content within a website. By simply clicking on the linked element, the user is directed to the desired website. (*see* 🗎 *210)*

Hub

A hub - often referred to as a star coupler - offers the possibility of connecting several network participants in a star configuration. Data packets received on one port are equally output on all other ports.

In addition to hubs for 10BaseT (10Mbit/s) and 100BaseT (100Mbit/s), there are so-called autosensing hubs that automatically detect whether the connected terminal device is operating at 10 or 100Mbit/s. Via autosensing hubs older 10BaseT devices can be integrated into new 100BaseT networks without any problems. (*see* 🗎 *15)*

ICMP – Internet Control Message Protocol

The ICMP protocol is used to transfer status information and error messages between IP network nodes. ICMP also offers the possibility of an echo request (see Ping); this way it can be determined whether a destination can be reached. (*see* 🗎 *73)*

Internet

The internet is the world's largest network connection, providing connected network participants with an almost limitless communication infrastructure. By using TCP/IP, the network participants can make use of services offered on the internet such as e-mail, FTP, HTTP etc. independently of the platform.

Intranet

A closed network (e.g. within a company), within the limits of which the network participants can make use of typical internet services such as e-mail, FTP, HTTP, etc. Usually there are also transitions from an intranet to the internet via routers or firewalls.

IP – Internet Protocol

Protocol that enables the connection of participants that are positioned in different networks. (*see* 🗎 *24ff)*

IP Address

The IP address is a 32-bit number that uniquely identifies each network participant in the internet or intranet. It consists of a network part (Net-ID) and a user part (Host-ID). *(see 🗎 25)*

IPX

Stands for Internet Packet Exchange and was developed by Novell as a network protocol for Novell-Netware.

IPsec

IPsec is a protocol that connects local networks securely and encrypted over public networks such as the internet. IPsec is used for setting up VPNs (Virtual Private Networks). *(see 🗎 191)*

ISDN – Integrated Services Digital Network

ISDN was introduced in the 1980s as a new standard in telecommunications. ISDN integrates telephone and fax, but also video telephony and data transmission. Thus, depending on the respective terminal equipment, voice, text, graphics and other data can be transmitted via ISDN.

ISDN provides two basic channels (B channels) each with 64 Kbit/s and one control channel (D channel) with 16 Kbit/s via the S0 interface of a basic access. The digital subscriber line has a combined maximum transmission speed of 144 Kbit/s (2B+D). In the two B-channels, two different services with a bit rate of 64 Kbit/s can be served simultaneously via one line. *(see 🗎 1198)*

ISDN-Router

ISDN routers allow two local networks to be connected via the ISDN network of a telephone network provider. In addition to the normal functions of a router, ISDN routers also handle the ISDN connection.

JSON

Markup language whose syntax is based on JavaScript. *(see 🗎 119)*

L2TP - Layer 2 Tunneling Protocol

With the L2TP protocol, data can be tunneled between two networks without encryption. *(see 🗎 193)*

LAN – Local Area Network

Local network within a limited area using a fast transmission medium such as Ethernet.

LTE - Long Term Evolution

In connection with LTE, there is often talk of the fourth generation of mobile telephony, which is not entirely correct. LTE is a purely IP-based data network. For telephony, LTE-enabled terminals currently still use the completely normal GSM/UMTS network.

LTE is intended to create the first uniform international mobile communications standard. LTE is therefore flexible in terms of the radio frequencies used. Two frequency ranges are used in Germany:

- 800MHz
- 2,6GHz

The 800MHz frequency band was formerly used for the transmission of analog television channels and became free when this technology was no longer used. A big advantage of this frequency band is the long range of up to 30km. This means that rural areas can also be well covered by LTE. The 2.6 GHz frequency band is primarily used in conurbations with smaller radio cells. A more efficient coding method and significantly improved technology on the provider side allow transfer rates of up to 100Mbit/s (theoretically even 300Mbit/s) for downloads and 50Mbit/s (theoretically even 100Mbit/s) for uploads.

Thanks to the high transmission speeds, LTE is a good alternative to a DSL connection. However, as with other mobile communications technologies, the total bandwidth available in a radio cell is divided among the number of active users. (*see* 📄 *202ff)*

MAC-ID

The unchangeable physical address of a network component. MAC = Media Access Control. (See Ethernet address)

NAT – Network Address Translation

Due to the explosive expansion of the internet in recent years, free IP addresses have become scarce and are only allocated very sparingly. NAT is used where company networks are connected to the internet. The company network is connected to the internet via a NAT-capable router, but internally it works with its own IP address space that is independent of the internet. From the outside, the network can only be addressed via a single (or a few) IP address(es). The port number in the received TCP/IP packet is used to route the packet to a specific internal network node.

Network node

All end devices connected to the network can also be called network nodes.

NTBA

Network Termination for ISDN Basic rate Access, in short NTBA, was the line termination for ISDN connections.

Ping – Packet Internet Gopher

Ping is used in TCP/IP networks for diagnostic purposes. This function can be used to check whether a particular station exists in the network and is actually addressable.

The ICMP packets used by Ping are defined in the internet standard RFC-792.
(see 🗎 73)

PoE - Power over Ethernet

Using PoE, Ethernet end devices can draw power from the network cable and thus do without an additional power supply. The supply voltage is fed into the network cable by special PoE switches or special intermediate adapters. *(see 🗎 16)*

POP3 – Post Office Protocol Version 3

To retrieve incoming e-mails from the mailbox on the mail server, the POP3 protocol is used in most cases. POP3 is based on TCP.
(see 🗎 107)

Port

When TCP and UDP are used the port number determines to which application an incoming data packet is forwarded. *(see 🗎 29)*

PPP – Point to Point Protocol

PPP is an extended successor to SLIP and features improved error correction, among other things.

Just like SLIP, PPP offers the possibility to connect TCP/IP devices that do not have a LAN connection to TCP/IP networks via the serial interface.
(see 🗎 56)

PPS-System - Production planning system

Software solution for production planning with the aim of reducing production times, optimizing inventory and storage quantities and meeting deadlines.

The best known provider of PPS systems is SAP.

PPTP - Point-to-Point Tunneling Protocol

PPTP was originally developed by Microsoft, 3COM and other companies to connect PCs to servers via public networks. Since PPTP is a component of Windows operating systems, it is still used today for setting up VPN. (*see* 🗎 *190)*

Proxy (-Server)

A proxy is a server that temporarily stores the contents of web pages. If a proxy is available, the browser requests the desired web page from the proxy, specifying the actual URL. If the contents are already cached there, they are not reloaded from the internet, but taken from the internal memory and passed on to the browser. This reduces data traffic to the internet. Content stored by the proxy is deleted or reloaded after a certain period of time to ensure that the database remains up-to-date. (*see* 🗎 *100)*

Repeater

In 10Base2 networks, repeaters are used to connect two Ethernet segments to extend the network beyond the reach of a single segment. Repeaters pass data packets from one network segment to another by “refreshing” the electrical signals in accordance with current standards, but leaving the content of the data packets unchanged. If the repeater detects a physical error on one of the connected segments, the connection to this segment is disconnected (“partitioned”). The partitioning is automatically canceled when the error is no longer present.

There must not be more than four repeaters between two stations. However, this rule only applies to repeaters located “one behind the other” - so a large number of repeaters can be used when implementing tree-like network structures.

RIP – Routing Information Protocol

Routing protocols such as RIP are used to forward changes in the routes between two networked systems to the systems involved, thus enabling dynamic changes in the routing tables. RIP is defined in the internet standard RFC-1058.

Router

Routers connect two different networks, whereby, in contrast to bridges, it is not the Ethernet address but the IP address that determines which data packets are to be forwarded to where. (*see* 🗎 *39)*

SLIP – Serial Line Internet Protocol

SLIP offers an easy way to transmit TCP/IP data packets over serial point-to-point connections. This means that end devices that do not have a LAN connection can also be integrated into the network via the serial interface.

SLIP works according to a very simple algorithm without its own data backup procedures: The actual IP data packet is preceded by a start character (decimal 192) and an end character (also decimal 192). To maintain binary transparency, start and end characters occurring in the data packet are first replaced by other sequences. SLIP is described in RFC 1055.
(see 🗎 55)

SLIP-Router

A SLIP router provides the hardware and functionality to integrate serial end devices with a TCP/IP stack into a network.

Com servers provide e.g. SLIP routing as operating mode.

SMTP – Simple Mail Transfer Protocol

SMTP controls the sending of e-mails from the mail client to the mail server (SMTP server) and between the mail servers and is based on TCP. *(see 🗎 106)*

SNMP – Simple Network Management Protocol

SNMP is based on UDP and enables the central administration and monitoring of network components.

SNMP is specified in the following standards: RFC 1052, RFC 1155, RFC 1156, RFC 1157, RFC 1213 and RFC 1441. *(see 🗎 85)*

STP – Shielded Twisted Pair

Shielded data cable in which 2 cable wires are twisted together. (See Twisted Pair)

Subnet-Mask

32-bit value that determines which part of the IP address addresses the network and which addresses the network node. *(see 🗎 37)*

Switch

Like a hub, a switch offers the possibility of connecting several network participants in a star configuration. Switches combine the functionality of a hub with that of a bridge: A switch “learns” the Ethernet address of the network node connected to a port and forwards only those data packets that are addressed to this network node.

An exception to this are broadcast messages which are forwarded to all ports (here the switch differs in its function from a bridge which generally does not forward broadcast messages). *(see 🗎 15)*

TCP – Transmission Control Protocol

TCP is based on IP and not only ensures that the participants are connected during data transmission, but also ensures the correct delivery of data and the correct sequence of data packets. *(see 🗎 28)*

TCP/IP-Stack

Part of the operating system or a driver attached to the operating system that provides all functions and drivers required for IP protocol support.

Telnet – Terminal over Network

In the past, Telnet was mainly used for remote access over the network on UNIX servers. Using a Telnet application (Telnet client), remote access to another computer (Telnet server) can be made from any computer in the network. Today, Telnet is also used to configure network components such as Com servers. Under TCP/IP, Telnet is normally accessed via port number 23; however, other port numbers can be used for special applications. Telnet is based on TCP/IP as transmission and backup protocol.

Telnet is defined in the internet standard RFC 854. *(see 🗎 75)*

Terminating resistor

With coaxial network topologies such as 10Base5 or 10Base2, each network strand must be terminated with a terminator at the beginning and end. The value of the terminating resistor must correspond to the cable impedance; for 10Base5 or 10Base2 this is 50 Ohm.

TFTP – Trivial File Transfer Protocol

The Trivial File Transfer Protocol (TFTP) is another protocol besides FTP for transferring entire files. TFTP offers only a minimum of commands, does not support complex security mechanisms and uses UDP as a transfer protocol. Since UDP is an unsecured protocol, TFTP has implemented its own minimal security mechanisms. (See 🗎 83)

The Trivial File Transfer Protocol is described in the RFC standards 783, 906, 1350 and 1782 to 1785.

Transceiver

The word transceiver is a composition of transmitter and receiver. The transceiver implements the physical network access of a station to the Ethernet and is integrated on the network card in the modern Ethernet topologies 10Base2 and 10BaseT. Only with 10Base5 (see AUI connection) the transceiver is attached directly to the network cable as an external component.

Twisted Pair

Data cable in which two cable wires are twisted together. This significantly reduces crosstalk between the two wires in a cable. Twisted pair cables are divided into unshielded UTP cables (Unshielded Twisted Pair) and shielded STP cables (Shielded Twisted Pair).

TP cables are mainly used in network technology and are categorized according to their maximum transmission frequencies; in practice, two types are usually used today:

- Category 3 cables allow a maximum transmission frequency of 25MHz, sufficient for use in 10BaseT, but also 100BaseT4 networks.
- Category 5 cables allow a maximum transmission frequency of 100MHz, sufficient for all current network topologies.

UDP – User Datagram Protocol

UDP is a protocol that is based on IP like TCP, but in contrast to it works connectionless and has no security mechanisms. The advantage of UDP over TCP is the higher transmission speed. (*see* 🗎 *31)*

UMTS - Universal Mobile Telecommunications System

With UMTS, the third generation of mobile communications technology has emerged after analog mobile telephony and GSM. With UMTS, the focus is no longer on telephony. Rather, UMTS was already geared towards the use of a wide range of multimedia services during its development.

UMTS terminals transmit over a frequency band from 1920MHz to 1980MHz and receive at 2110MHz to 2170MHz. The usable individual frequencies are 5MHz apart. Several hundred channels can be operated on a single frequency. This simultaneous use of a frequency is not regulated by fixed time allocation as with GSM. With UMTS, a special protocol regulates the use. Thus, a few users can transmit large amounts of data on one frequency or the frequency can be used by many users to transmit smaller amounts of data.

In this way transmission rates of up to 384Kbit/s can be achieved.
(see 🗎 203)

URL – Uniform Resource Locator
Address and protocol information for the browser. The user uses the URL to tell the browser which protocol is used, on which web server the page is located, and where it can be found on the web server. *(see 🗎 206)*

UTP – Unshielded Twisted Pair
In contrast to Shielded Twisted Pair an unshielded data cable where two cable wires are twisted together.

VPN - Virtual Private Network
VPN describes the technique of connecting confidential network parts at different locations via the internet, i.e. a public network.
(see 🗎 182)

Web-Based Management
Web-based management is the ability to configure end devices over the network directly in the browser window without special software.

Web-IO
Small boxes with Ethernet connection and integrated web server. Web-IO can make digital or analog signals accessible via TCP/IP Ethernet or visualize or control them in a browser.
(see https://www.wut.de/web-io)

Wireless LAN
WLAN realizes the network connection via radio. *(see 🗎 19)*

WWW – World Wide Web
WWW is often equated with the internet. This is not quite true: While the internet describes the physical connection paths, the WWW defines the entirety of web pages or documents linked via the internet, which can be loaded and displayed by the browser via the HTTP protocol.

Number systems

When we deal with numbers in everyday life, these are usually decimal numbers. We are familiar with the decimal number system and even as a child everyone learns that after the nine comes the ten and the written number therefore has one more digit.

Computers have a different way of dealing with numbers than humans and therefore we want to shed some light on the background of number systems.

Value and display

As a human being, we deal with numbers and calculations on a daily basis - they are written down in an Math booklet, They are number of coins in a pocket or they are printed on traffic signs, etc.

The human brain processes all of this as numerical values and can calculate immediately.

Computers, on the other hand, make a strict distinction between numerical values that can be used for calculations and numbers or figures that are displayed on a screen, for example.

Anyone who has ever dealt with programming languages knows that “written down” numbers must first be converted into values by the computer before it can calculate with them.

In computer technology we also have to deal with different number systems.

- Decimal number system
- Dual/Binary number system
- Hexadecimal number system

When we talk about different number systems, we also talk about the fact that one and the same value can be represented or written down in different ways.

The Decimal System

In order to understand other number systems, it is important to first understand why the decimal number system is the way it is.

The decimal number system is based on the base 10, which is because humans have ten fingers (which can be used for counting).

To represent decimal numbers, a character set from 0 to 9, i.e. 10 characters, is available.

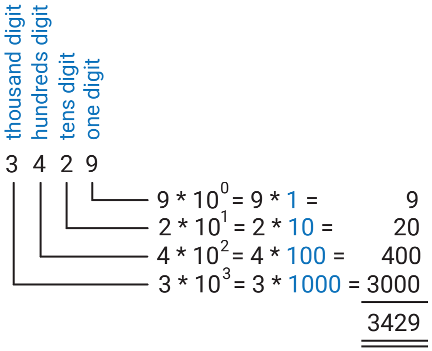
Numbers greater than nine are given a new digit, the tens digit. Numbers greater than 99 are given the hundreds digit.

Let’s take a look at an example of the systematics behind this:

Mathematically speaking, each digit represents the corresponding power of ten multiplied by the digit in that position.

As a reminder of mathematics lessons:

- each number to the power of 0 is equal to 1
- each number to the power of 1 is the number itself
- each number to the power of 2 is the number multiplied by itself
- each number to the power of 3 is the number multiplied by itself and the result multiplied again by the number
- and so on



$$
\begin{aligned}
9 * 10^{0} &= 9 * 1 = & 9 \\
2 * 10^{1} &= 2 * 10 = & 20 \\
4 * 10^{2} &= 4 * 100 = & 400 \\
3 * 10^{3} &= 3 * 1000 = & 3000 \\
& & \underline{\underline{3429}}
\end{aligned}
$$

The dual/binary number system

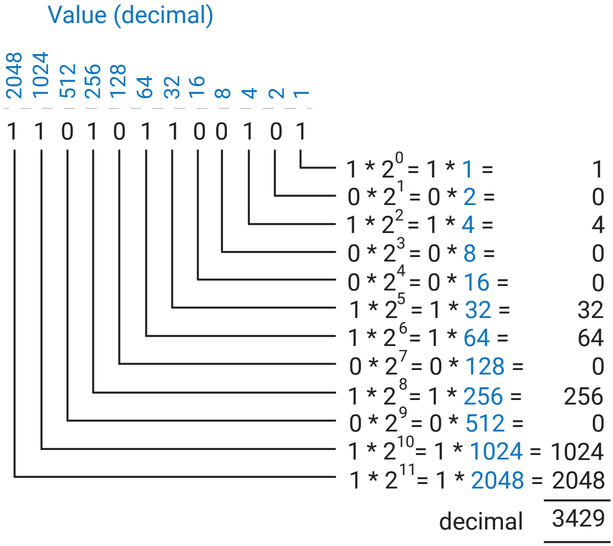
The dual number system, often referred to as binary number system, is the number system with which microprocessors and thus computers work internally. It knows only two digits, namely 0 and 1, to represent a number.

As a result, numbers greater than 1 already have an additional digit. If you count up dual numbers, the result is the following:

```
  0
  1
 10
 11
100
...
```

The systematics behind it is the same as in the decimal number system, only that it is based on powers of two instead of powers of ten.

Our number 3429 looks like this, written in dual notation: 110101100101



But why do microprocessors actually work with dual numbers? This is due to the fact that microprocessors only know two states at the lowest level: ON and OFF or 1 and 0. One digit in the dual system is one bit.

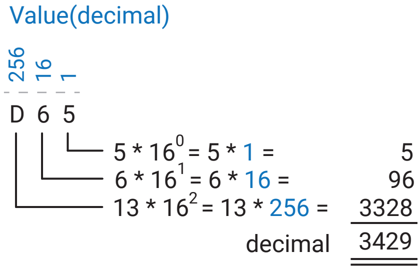
The hexadecimal number system

With the hexadecimal number system there are 16 possible digits or characters within one digit. Since there are only 0 to 9 as digits, values greater than 9 are represented by the letters A to F (A=10, B=11, C=12, D=13, E=14 and F=15).

If you count up hexadecimal numbers, it looks like this:

```
 .
 8
 9
 A
 B
 C
 D
 E
 F
10
11
..
```

And again the system is the same as in the decimal and dual number system. However, the hexadecimal number system uses powers on the base 16:



Now one can rightly ask: Why do we need another number system that seems even more complicated and unmanageable than the dual system?

As already explained, computers work with bits and bytes - in some cases even with 16-bit, 32-bit or 64-bit values.

From a logical point of view, the dual number system is best suited to represent such values, since each bit is represented by one digit. For humans, however, columns of ones and zeros are difficult to comprehend.

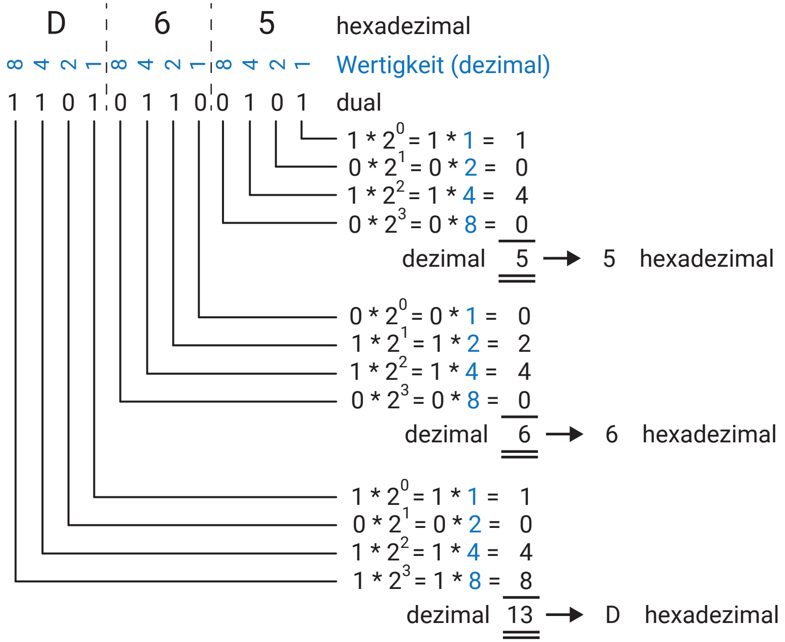
For example, if a few 16-bit values are to be entered, the probability of typing is quite high.

Now, of course, you could enter the values in decimal notation, but in doing so, any reference to the bit pattern of the dual number is lost.

The fact that 3429 decimal = 110101100101 is dual cannot be seen without a complex conversion. And this is where the hexadecimal number system comes into play.

It is no coincidence that four digits dual always correspond exactly to one digit hexadecimal. The hexadecimal number system works with powers to the base 16 and 16 is again the result of a power of two, which is 2^4.

This means that only four digits at a time must be converted dual - i.e. four bits.



With a little practice you can convert dual numbers in your head to hexadecimal and vice versa.