W&T

www.WuT.de

Frank Thiel

TCP/IP-Ethernet

Mach es einfach. Netzwerktechnische Grundlagen.

Vorwort

W&T verbindet

Wir blicken zurück auf über 40 Jahre Erfahrung in der Entwicklung und Produktion von Mikrocomputertechnik. Mit der Zeit kamen Netzwerk- und Sensortechnik, sowie IT-Sicherheitskomponenten hinzu. Unsere blauen Boxen machen Ihre Geräte, Schalt- und Steuersignale sowie Sensordaten sicher und zuverlässig in Ihrem Netzwerk verfügbar.

www.wut.de

Netzwerktechnisches Grundwissen finden Sie seit 15 Jahren in diesem mittlerweile etablierten, in der achten überarbeiteten und erweiterten Auflage gedruckten Grundlagenbuch.

Mit diesem Büchlein hoffen wir, Ihnen die nötige technische Orientierung zu geben. Sie werden sehen: Hier finden Sie Kompliziertes einfach erklärt.

Also nur Mut und viel Freude beim Entdecken!

Entdeckerfreude wünschen wir uns übrigens auch für unsere Kinder.
Deshalb unterstützen wir die Arbeit der Winzig Stiftung im Bereich der frühen Kindheit.

www.winzig-stiftung.de

Mit den besten Wünschen für die Lektüre des Büchleins,
für das Gelingen Ihrer Projekte
und noch viele gute verbundene Jahre

Ihr Rüdiger Theis, Frank Thiel
und alle WuTler

Rüdiger Theis

Frank Thiel

Sie haben Fragen zu unseren Produkten oder konkreten Anwendungsfällen?

Unsere Techniker sind gerne für Sie da: +49 (0) 202 2680-110

Inhalt

Einführung

Computer- bzw. Datennetzwerke erlauben einer unbestimmten Anzahl an Netzwerkteilnehmern, über eine gemeinsame Infrastruktur beliebig Daten miteinander auszutauschen.

Kommunikationsdaten

Wenn wir von Datennetzwerken sprechen, gilt es zunächst zu definieren: Was sind eigentlich Daten und in welcher Form werden sie kodiert?

Egal ob Text, Webseiten, Bilder, Musik, Videos oder andere Daten übertragen werden sollen - es wird immer eine bestimmte Menge an Bytes von A nach B transportiert.

Bits und Bytes

Auf unterster Ebene arbeiten Computer mit Bits, also mit Speicherstellen, die den Wert 1 oder 0 haben können. Acht Bits bilden ein Byte.

Ein Byte ist ein Zahlenwert zwischen 0 und 255. In der Datentechnik werden Bytes für gewöhnlich in zweistelliger hexadezimaler Schreibweise dargestellt - also 00 bis FF (siehe Kapitel Zahlensysteme).

Kodierung

Je nach Anwendung wird z.B. aus einem Text eine bestimmte Menge an Bytes, wobei jedes Byte einem Buchstaben entspricht. Die Zuordnung, welches Schriftzeichen welchem Zahlenwert entspricht, ist in der ASCII-Tabelle definiert (ASCII = American Standard Code for Information Interchange).

Bei einem Bild wäre in einer Menge an Bytes codiert, welcher Bildpunkt an welcher Position welche Farbe hat.

Welche Bedeutung die einzelnen Bytes in der Anwendung haben, spielt für den Transport keine Rolle. Hier ist es nur eine entsprechende Menge an Bytes, also Zahlen, mit denen man bei Bedarf Rechenoperationen vornehmen kann.

Funktionsprinzip Netzwerk

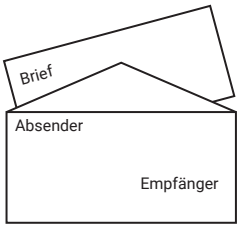
Grundsätzlich haben alle Netzwerktopologien eines gemeinsam:

Jeder Netzteilnehmer erhält (mindestens) eine eigene und eindeutige Adresse.

Die zu übertragenden Nutzdaten werden in einen Rahmen aus z. B. Adresse des Empfängers, Adresse des Absenders und Checksumme in ein „Datenpaket" verpackt. Zusammengefasst bezeichnet man das Regelwerk, nach dem ein solcher Rahmen aufgebaut ist, auch als Netzwerkprotokoll bzw. Protokoll.

Mit Hilfe der Adressinformationen können die Nutzdaten in den so entstandenen Datenpaketen über gemeinsam benutzte Leitungswege an den richtigen Empfänger übermittelt werden.

Bei einem Brief ist es nicht anders: Man steckt den Brief in einen Umschlag, auf dem Empfänger und Absender notiert sind. Der Postbote weiß dann, wem er den Brief zustellen soll; der Empfänger kann ablesen, woher er kommt und wem er bei Bedarf antworten kann.



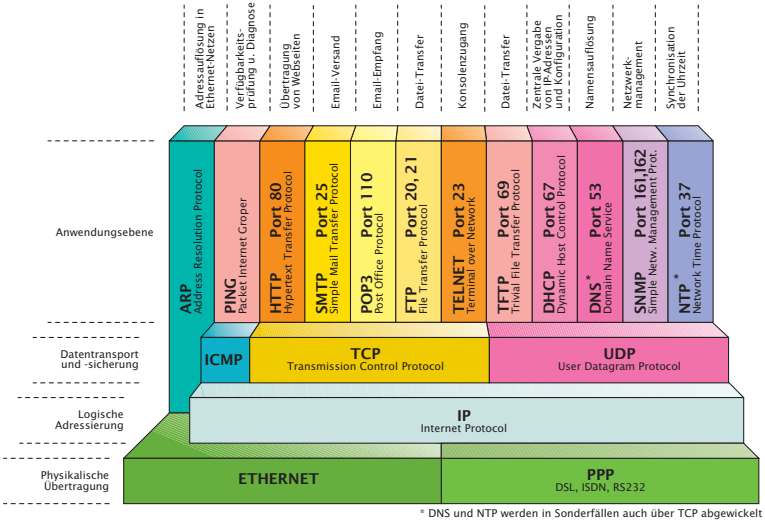
Beim Datentransfer innerhalb eines Netzwerkes hat der Empfänger zusätzlich die Möglichkeit, mit Hilfe der mitversandten Checksumme die Vollständigkeit und Fehlerfreiheit der empfangenen Nutzdaten zu überprüfen.

Auf ihrem Weg von einer Anwendung zur anderen durchlaufen die Daten verschiedene Protokollschichten. Jede dieser Schichten übernimmt dabei eine andere Funktion, auf die die nächst höhere Schicht wiederum aufbaut.

Die unterste Schicht ist der physikalische Netzzugang. In lokalen Netzen sind hier die verschiedenen Ethernet-Standards üblich. Wir werden später noch sehen, wie in den Datenpaketen der untersten Schicht tatsächlich auch alle Informationen für die höheren Schichten mit übermittelt werden.

Soll das Ethernet-Datenpaket in ein fremdes Netz versandt werden, wird es von übergeordneten Protokollen, z.B. TCP/IP, adressiert und transportiert.

TCP/IP liefert das Datenpaket nicht nur beim richtigen Empfänger ab, sondern auch bei der richtigen Applikation. Dazu wird in aller Regel ein weiteres übergeordnetes Protokoll verwendet, welches mit dem entsprechenden Anwendungsprogramm zusammenarbeitet. Sie erhalten z.B. eine E-Mail über das Protokoll POP3 und können diese mit Ihrem E-Mail-Programm abrufen.



* DNS und NTP werden in Sonderfällen auch über TCP abgewickelt

Physikalische Übertragung

Je nach Anwendungsbereich stehen verschiedene physikalische Vernetzungstechnologien zur Verfügung. Bei lokalen Netzwerken ist Ethernet der heute am meisten verbreitete Netzwerkstandard; bereits 1996 waren ca. 86% aller bestehenden Netzwerke in dieser Technologie realisiert. Zu dieser Zeit wurden Netzwerke fast ausschließlich im Büroumfeld genutzt. Inzwischen hat sich Ethernet an vielen Stellen auch im industriellen Umfeld durchgesetzt und verdrängt in den Werkhallen zunehmend die bisher üblichen Übertragungsverfahren wie z.B. serielle Feldbusse.

Lokale Netze mit Ethernet

Ethernet in seiner ursprünglichen Form ist in der IEEE-Norm 802.3 standardisiert. Vereinfacht gesagt überträgt Ethernet mit Hilfe verschiedener Algorithmen Daten in standardisierten Paketen über ein Medium an die Teilnehmer des Netzes. Dabei hat jeder Teilnehmer eine eindeutige Adresse.

Ursprüngliche Ethernet-Standards

Im Laufe der Zeit haben sich verschiedene Ethernet-Varianten herausgebildet, die sich maßgeblich in der Übertragungsgeschwindigkeit und den verwendeten Kabeltypen unterscheiden lassen. Ethernet wurde ursprünglich mit einer Übertragungsgeschwindigkeit von 10 Mbit/s betrieben; hierbei gab es drei verschiedene Grundmodelle, die heute keine Bedeutung mehr haben und allenfalls noch in Altinstallationen vorhanden sind:

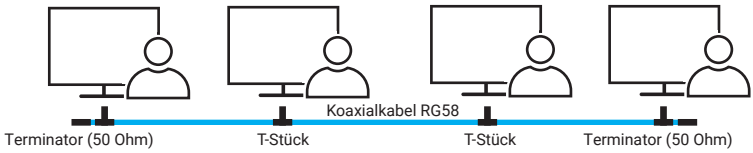
10Base5

Auch oft als „Yellow Cable“ bezeichnet; stellt den ursprünglichen Ethernet-Standard dar und hat heute keine Bedeutung mehr. Verwendet wurde ein fingerdickes, unflexibles und meist gelbes Koaxialkabel; die Reichweite betrug 500m.

10Base2

10Base2 wird heute bei Neuinstallationen nicht mehr verwendet und ist nur noch recht selten in älteren Netzwerkinstallationen zu finden.

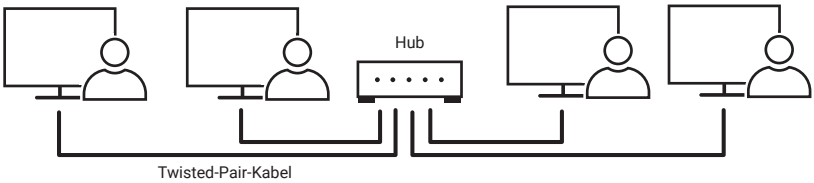
10Base2 ist auch bekannt als Thin Ethernet, Cheapernet oder schlicht als BNC-Netzwerk. Alle Netzteilnehmer werden parallel auf ein Koaxialkabel (RG58, 50 Ohm Wellenwiderstand) aufgeschaltet. Das Kabel muss an beiden Enden mit einem 50-Ohm-Terminator (Endwiderstand) abgeschlossen sein.



Teilen sich mehrere Geräte einen gemeinsamen Leitungsweg, spricht man auch von einer Bustopologie. Der Nachteil dieser Technik liegt in der hohen Störanfälligkeit. Wird die RG58-Verkabelung an einer beliebigen Stelle unterbrochen, ist der Netzwerkzugriff für alle angeschlossenen Netzteilnehmer gestört.

10BaseT

Jeder Netzteilnehmer wird über ein eigenes Twisted-Pair-Kabel an einen sogenannten Hub (Sternverteiler) angeschlossen, der alle Datenpakete gleichermaßen an alle Netzteilnehmer weitergibt.



Auch wenn 10BaseT physikalisch sternförmig arbeitet, bleibt von der Logik her das Busprinzip erhalten, da alle angeschlossenen Netzwerkteilnehmer den gesamten Netzwerkverkehr empfangen.

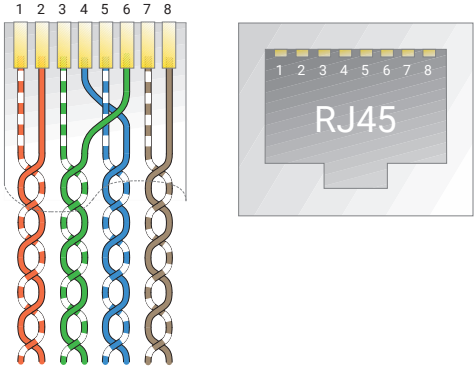
Aktuelle Ethernet-Standards

Aktuelle drahtgebundene Ethernet-Netzwerke verwenden so wie auch 10BaseT Twisted-Pair-Kabel. Die verwendeten Kabel kommen ursprünglich aus der amerikanischen Telefontechnik. Twisted-Pair bedeutet, dass die für jeweils ein Signal verwendeten Kabeladernpaare miteinander verdrillt sind. Gebräuchlich sind Kabel mit vier Adernpaaren.

Auch die verwendeten RJ45-Steckverbinder entstammen der amerikanischen Telefontechnik. Die zunächst etwas merkwürdig anmutende Aufteilung der einzelnen Paare und deren Farbgebung ist im AT&T Standard 258 festgeschrieben. 10BaseT benutzt nur die Pins 1 und 2, sowie 3 und 6.

Twisted-Pair-Kabel für die Netzwerktechnik sind entsprechend ihrer Übertragungs-

eigenschaften kategorisiert. Für 10BaseT wurden Kabel der Kategorie 3 (Cat 3) benötigt. Bei aktuellen Netzwerken kommen Kabel zum Einsatz, die mindestens der Kategorie 5 entsprechen. Die maximal zulässige Kabellänge zwischen zwei aktiven Komponenten liegt bei 100m.



100BaseT

Mit zunehmend größeren Datenmengen wurde in den 90er Jahren Fast Ethernet mit einer Übertragungsgeschwindigkeit von 100Mbit/s eingeführt.

Genau wie bei 10BaseT erfolgt die Verkabelung der Netzteilnehmer über Twisted-Pair-Kabel. Als Sternverteiler werden anstelle von Hubs Switches eingesetzt. Switches filtern den Datenverkehr, so dass jeder angeschlossene Teilnehmer nur die für ihn bestimmten Daten bekommt (mehr zu Switches auf der nächsten Seite). Die verwendeten Kabel müssen wie bereits angemerkt mindestens der Kategorie 5 (Cat. 5) entsprechen. Die maximale Kabellänge beträgt 100m.

1000BaseT - Gigabit-Ethernet

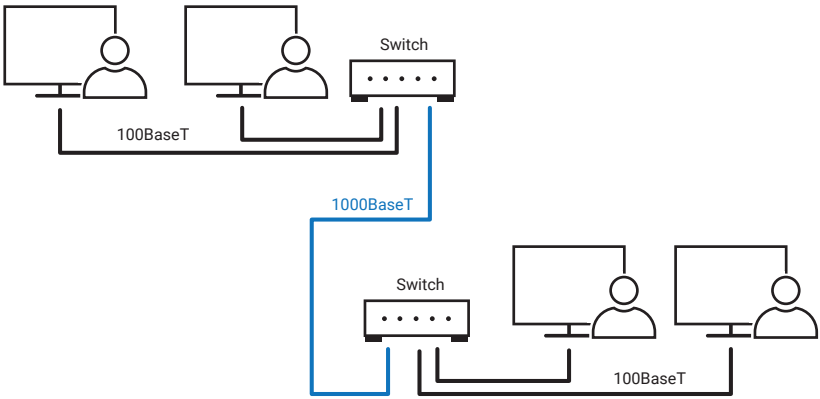
Der nächste Ethernet Standard, mit dem Übertragungsgeschwindigkeiten von einem Gigabit (1000 Megabit) pro Sekunde möglich sind, ist 1000BaseT. Um diese hohe Bitrate zu erreichen, arbeitet 1000BaseT mit einem speziellen Datenkodierungsverfahren.

Die Anforderungen an die Verkabelung sind die gleichen wie bei 100BaseT. Es werden allerdings alle vier Adernpaare der Twisted-Pair-Kabel parallel genutzt, die mindestens der Kategorie 5 entsprechen müssen. 1000BaseT kann über max. 100m betrieben werden.

Anfänglich wurde Gigabit-Ethernet vorwiegend als Hintergrundverkablung zwischen

Switches verwendet. Solche schnelleren übergeordneten Verbindungen werden auch als Backbone bezeichnet.

Heutige PCs haben im Allgemeinen einen Gigabit-Ethernet-Anschluss, so dass eine direkte, schnelle Anbindung nichts Außergewöhnliches mehr ist.



10GBaseT
Inzwischen sind über Twisted-Pair-Kabel Übertragungsraten von bis zu 10 Gigabit/s möglich. Die 10GBaseT-Technik bedarf allerdings spezieller Netzwerkkarten und Infrastrukturkomponenten und wird deshalb z.Zt. nur zur Direktverbindung von Servern oder als Backbone verwendet. Bei entsprechend hochwertigem Kabel (min. Cat. 6, besser Cat. 7) sind ebenfalls Distanzen bis zu 100m möglich.

Switch und Hub
Als sich 10BaseT als physikalischer Standard für Ethernet-Netzwerke durchgesetzt hatte, wurden zunächst nur Hubs als Sternverteiler eingesetzt. Hubs leiten, wie bereits beschrieben, den gesamten Datenverkehr des Netzwerkes an alle angeschlossenen Netzwerkteilnehmer weiter.

Als sich später 100BaseT als neuer Standard etabliert hatte, wurden die Hubs mit Autosensing-Ports ausgestattet. Autosensing bedeutet, dass der Ethernet-Anschluss automatisch erkennt, mit welcher Geschwindigkeit das angeschlossene Endgerät arbeitet. Beide beteiligten Schnittstellen einigen sich dann darauf, ob 10BaseT oder 100BaseT verwendet wird.

Inzwischen werden anstelle von Hubs ausschließlich Switches eingesetzt. Switches leiten nicht mehr den gesamten Ethernet-Datenverkehr an alle angeschlossenen

Netzwerkteilnehmer weiter. Stattdessen filtern Switches den Datenstrom so, dass am entsprechenden Port nur noch die Daten ausgegeben werden, die für den dort angeschlossenen Netzteilnehmer bestimmt sind.

Der Vorteil dieser Technik liegt darin, dass den einzelnen Anschlüssen die volle Bandbreite der Netzwerkanbindung zur Verfügung steht. Damit erhöht sich die Geschwindigkeit der Datenübertragung für die Netzwerkteilnehmer.

Die Autosensing-Fähigkeiten von Switches umfassen darüber hinaus heute meist auch 1000BaseT.

Spezielle physikalische Ethernet-Standards

Neben den bis hierhin vorgestellten herkömmlichen Verkabelungsvarianten gibt es inzwischen weitere Möglichkeiten, Teilnehmer an ein Netzwerk anzuschließen.

PoE - Power over Ethernet

Wenn von Netzwerkteilnehmern gesprochen wird, denken die meisten zunächst an einen PC. Jeder stationäre PC benötigt neben dem Netzwerkkabel zumindest ein weiteres Kabel zur Stromversorgung - meist 230V. Es gibt aber auch Netzwerkteilnehmer wie z.B. die W&T Web-Thermometer, die zum einen deutlich kleiner sind als ein PC und zum anderen mit relativ wenig Versorgungsenergie auskommen.

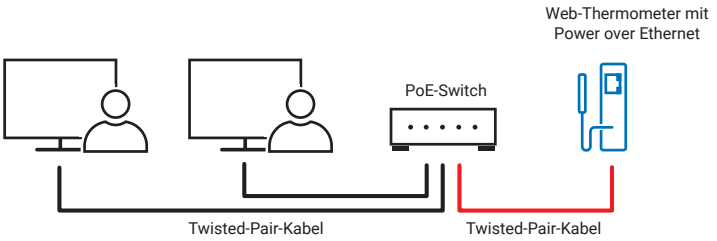
Mit PoE lassen sich solche Geräte über die ganz normale Ethernet-Verkabelung zusätzlich mit Strom versorgen. Damit das funktionieren kann, wurde die Ethernet-Schnittstelle dieser Geräte entsprechend technisch erweitert. Zum Betrieb sind außerdem spezielle Switches oder PoE-Injektoren nötig, welche die benötigte Energie in das Netzwerkkabel einspeisen.

PoE versorgt die Endgeräte mit 48V und kennt z.Zt. fünf verschiedene Leistungsklassen, die sich durch die max. aufgenommene Leistung unterscheiden. Durch ein besonderes Kodierungsverfahren erkennt der PoE-Switch, ob ein angestecktes Gerät PoE-fähig ist oder nicht, und schaltet die Versorgung nur bei Bedarf, bzw. wenn die benötigte Leistung auch zur Verfügung gestellt werden kann, ein.

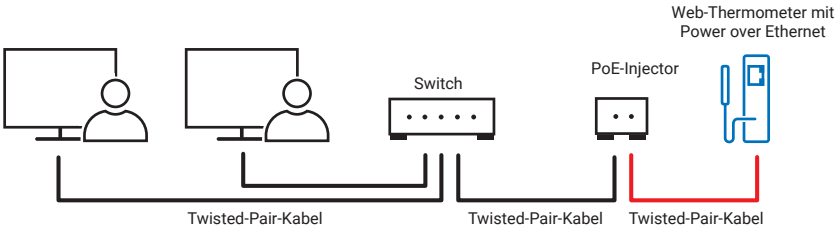
Klasse	Max. Speiseleistung	Entnahme-leistung	Beispiel für Endgeräte
0	15,4 W	0,44 W - 12,95 W	Individuelle Geräte

Klasse	Max. Speiseleistung	Entnahme-leistung	Beispiel für Endgeräte
1	4,0 W	0,44 W - 3,84 W	W&T Web-IO u. Com-Server
2	7,0 W	3,84 W - 8,49 W	IP-Telefonie
3	15,4 W	8,49 W - 12,95 W	Überwachungskamera
4	25,5 W	12,95–25,50 W	Panel PC

So können am selben Switch normale Ethernet-Komponenten und PoE-Geräte gemischt betrieben werden.



Wenn die PoE-Versorgung aus einem Switch kommt, spricht man von einer Endspan-Lösung. In bestehenden Netzwerken können PoE-Geräte aber auch mittels eines zwischengeschalteten PoE-Injektors mit Strom versorgt werden.



Diesen Fall nennt man Midspan-Lösung.

Ethernet über Lichtwellenleiter

Bei Kabellängen über 100 Meter oder in stark elektromagnetisch gestörtem Umfeld stößt die Übertragungstechnik von 100/1000BaseT und 10GBaseT an ihre Grenzen.

Bei der Datenübertragung über Lichtwellenleiter (kurz LWL) werden die Ethernet-Daten in Lichtsignale umgesetzt und über eine Glasfaser weitergeleitet. Das hat den Vorteil, dass es keine elektrisch leitende Verbindung über das LWL-Kabel gibt. Insbesondere bei gebäudeübergreifenden Verbindungen bietet die Übertragung per LWL einen optimalen Schutz vor Gewitterschäden.

Bei den Glasfasern unterscheidet man grundsätzlich zwei physikalische LWL-Typen:

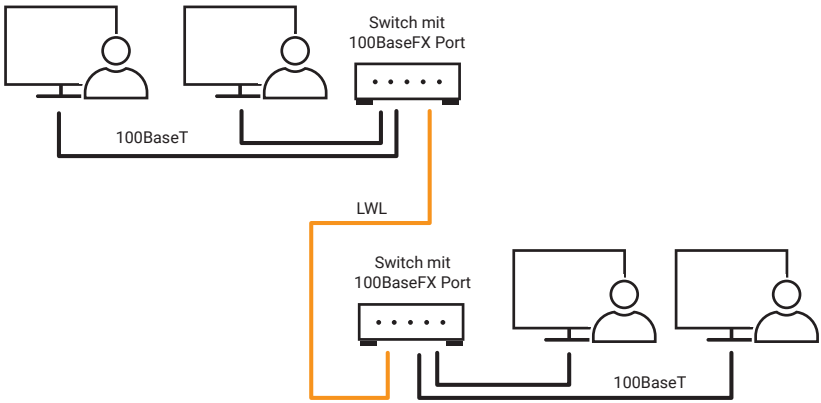
- **Multimodefasern**
  Mit Multimodefasern können Distanzen von bis zu 2 km überbrückt werden.

- **Monomodefasern**
  Eine ebenfalls gängige Bezeichnung für Monomode ist Singlemode. Die Verarbeitung und Konfektionierung von Monomodefasern ist deutlich aufwändiger als bei Multimodefasern. Dafür können aber je nach Übertragungssystem Entfernungen von bis zu 40 km überbrückt werden.

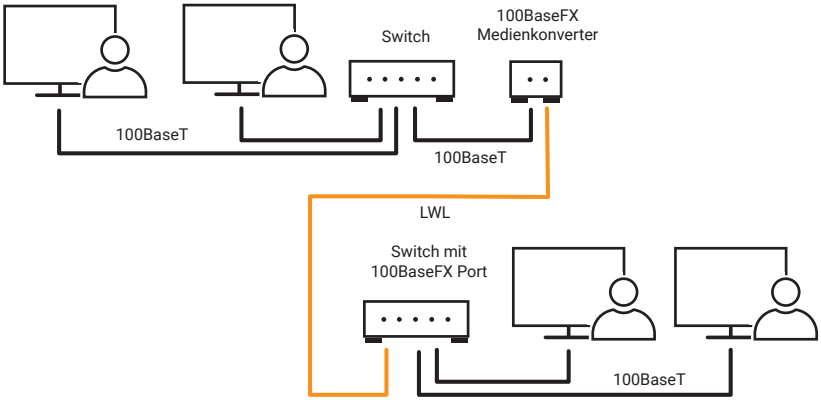
Eine detaillierte Beschreibung zu den unterschiedlichen LWL-Standards finden Sie im Netzwerk-ABC.

LWL-Topologien

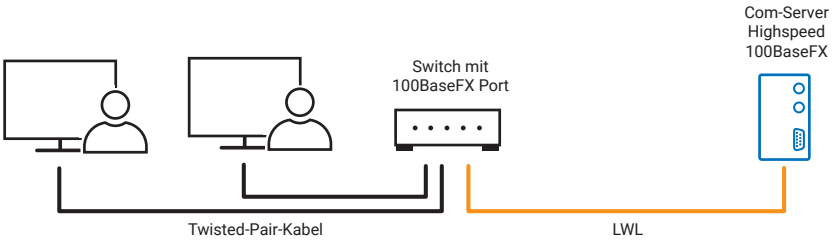
Da die Kosten für eine Glasfaser-Installation deutlich über denen von 100BaseT liegen, werden meist nur bestimmte Teile eines Netzwerkes als LWL ausgeführt. Zum Beispiel zwischen Switches als Backbone-Verbindung.



Wenn eine der verwendeten Komponenten von Hause aus keinen LWL-Anschluss hat, kann ein entsprechender Medienkonverter genutzt werden.



Es gibt aber auch Endgeräte, die bereits mit einem LWL-Port ausgestattet sind.



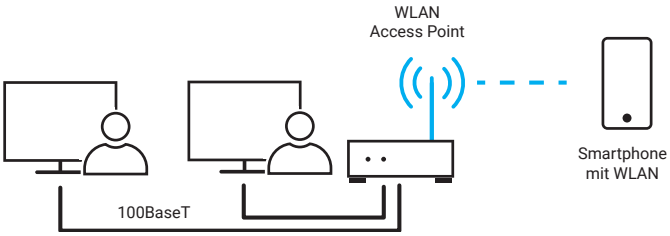
So zum Beispiel der W&T Com-Server Highspeed 100BaseFX.

Solche Lösungen bezeichnet man als „Fiber to the Desk“.

Wireless LAN

WLAN realisiert die Netzwerkanbindung über Funk und verschafft dem Nutzer damit Unabhängigkeit vom Kabel und somit Mobilität.

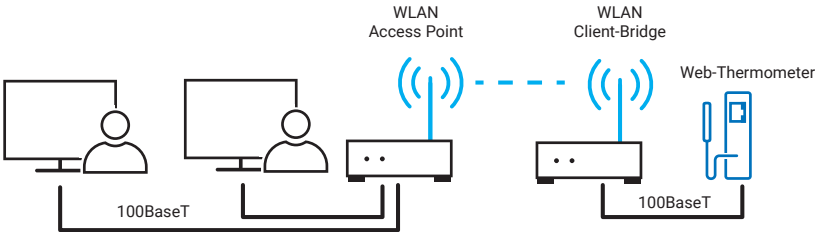
Im Allgemeinen besteht ein WLAN aus mindestens einem Access Point und einem WLAN-Client.



Der Access Point übernimmt die Rolle eines Sternverteilers. WLAN-Clients können sich beim Access Point anmelden und danach über Funk mit dem restlichen Netzwerk kommunizieren.

In den meisten Fällen sind Access Points in DSL-Routern oder Switches integriert und fungieren als Anbindung an ein kabelgebundenes Netzwerk.

Netzteilnehmer, die kein integriertes Wireless LAN Interface haben, können über eine WLAN-Client-Bridge Zugang zum WLAN bekommen. Die Client-Bridge fungiert als Medienkonverter zwischen funk- und drahtgebundenem Netzwerk.



Die Reichweite eines WLAN kann je nach Umgebung und eingesetzten Komponenten theoretisch bis zu 300 Meter betragen. Innerhalb von Gebäuden werden typische Werte von 25m angegeben, wobei Geschossdecken und Wände die Reichweite zusätzlich einschränken können.

Da sich die örtlichen Ausdehnungen von Funknetzwerken überschneiden können, gibt es mehrere mögliche Kanäle (Übertragungsfrequenzen). Bei mehreren WLANs an einem Standort (in Mehrfamilienhäusern oder Geschäftsgebäuden keine Seltenheit) sollte, wenn möglich, zwischen zwei benutzten Kanälen ein ungenutzter Kanal liegen, damit es nicht zu gegenseitigen Störungen kommt.

Ein weiterer Aspekt bei WLAN ist die Datensicherheit. Funksignale sind bei entspre-

chender technischer Ausstattung für jeden, der sich in Reichweite des WLAN befindet, empfangbar.

Um Funknetzwerke vor Fremdnutzung und „Mithören“ zu schützen, werden die Daten verschlüsselt. Ein regulärer WLAN-Teilnehmer muss sowohl das benutzte Verschlüsselungsverfahren als auch den richtigen Schlüssel anwenden, um Zugang zum Funknetz zu bekommen.

Ethernet-Standards im Überblick

Ethernet-Standard	Übertragungsmedium	max. Distanz	Datenrate	
10Base2	50 Ohm Koaxialkabel	185 m	10 MBit/s	*
10Base5	50 Ohm Koaxialkabel	500 m	10 MBit/s	*
10BaseT	100 Ohm TP-Kabel Kat.3	100 m	10 MBit/s	
100BaseT	100 Ohm TP-Kabel Kat.5	100 m	100 MBit/s	
1000BaseT/Gigabit	100 Ohm TP-Kabel Kat.5	100 m	1000 MBit/s	
10GBaseT	100 Ohm TP-Kabel Kat. 6 u. 7.	100 m	10 GBits/s	
100BaseT-PoE	100 Ohm TP-Kabel Kat.5	100 m	100 MBit/s	
1000BaseT-PoEt	100 Ohm TP-Kabel Kat.5	100 m	1000 MBit/s	
100BaseFX	Multimode LWL	2000 m	100 MBit/s	
1000BaseSX	Multimode LWL	550 m - 1000 m	1000 MBit/s	
1000BaseLX	Multimode LWL Singlemode LWL	550 m 5 km	1000 MBit/s	
WLAN 802.11a	Funk 5GHz	typisch 25 m	max. 54Mbit/s	*
WLAN 802.11b	Funk 2,4GHz	typisch 25 m	max. 11Mbit/s	*
WLAN 802.11g	Funk 2,4GHz	typisch 25m	max. 54 Mbit/s	*
WLAN 802.11n	Funk 2,4GHz und 5GHz	typisch 25m	max. 600 Mbit/s	*
WLAN 802.11ac	Funk 25GHz	typisch 25m	max. ca. 6900 Mbit/s	*

** Hier müssen sich die Netzwerkteilnehmer die maximale Datenrate teilen. Bei den anderen Standards steht die angegebene Datenrate jedem Netzteilnehmer zur Verfügung, wenn dieser über einen Switch mit dem Netzwerk verbunden ist.*

In der Tabelle sind nur die wichtigsten für lokale Netzwerke geeigneten LWL-Standards aufgeführt. Eine Gesamtübersicht findet sich im Netzwerk-ABC.

Verschiedene Ethernet-Standards kombinieren
Alle Ethernet-Standards lassen sich mit Hilfe entsprechender Infrastrukturkomponenten kombinieren bzw. mischen.

So können z.B. verschiedene Gebäudeteile über eine Glasfaserverkabelung miteinander verbunden werden. Entsprechende Switches übernehmen die Umsetzung auf 100BaseT oder Gigabit und können bei Bedarf sogar die PoE-Versorgung liefern. WLAN-fähige Geräte können über einen Access Point oder WLAN-Router an das Netzwerk angebunden werden.

Das Ethernet-Datenformat

Welches physikalische Grundmodell auch genutzt wird – der logische Aufbau der verwendeten Datenpakete ist bei allen Ethernet-Topologien gleich.

Die Ethernet-Adresse

Die Ethernet-Adresse, auch MAC-ID oder Node-Number genannt, wird vom Hersteller in den physikalischen Ethernetadapter (Netzwerkkarte, Printserver, Com-Server, Router ...) fest „eingebrannt“, steht also für jedes Endgerät fest und kann nicht geändert werden. Die Ethernet-Adresse ist ein 6-Byte-Wert, der üblicherweise in hexadezimaler Schreibweise angegeben wird. Eine Ethernet-Adresse sieht typischerweise so aus: 00-C0-3D-08-27-8B.

Die ersten drei Hex-Werte bezeichnen dabei den Herstellercode, die letzten drei Hex-Werte werden vom Hersteller meist fortlaufend vergeben.

Das Ethernet-Datenpaket

Es gibt vier verschiedene Typen von Ethernet-Datenpaketen, die je nach Anwendung eingesetzt werden:

Datenpakettyp	Anwendung
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

In Verbindung mit TCP/IP werden in aller Regel Ethernet-Datenpakete vom Typ Ethernet II verwendet.

Hier der Aufbau eines Ethernet-II-Datenpakets:

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

Preamble	Die Bitfolge mit stetigem Wechsel zwischen 0 und 1 dient zur Erkennung des Paketanfangs bzw. der Synchronisation. Eine Kollision (überschneidendes Senden zweier Teilnehmer) kann an einer gestörten Preamble erkannt werden. Das Ende der Preamble wird durch die Bitfolge „11“ gekennzeichnet.
Destination	Ethernet-Adresse des Empfängers
Source	Ethernet-Adresse des Absenders
Type	Gibt den übergeordneten Verwendungszweck an (z.B. IP = Internet Protocol = 0800h)
Data Bytes	Nutzdaten
FCS	Frame Checksum - Checksumme des Datenpakets

Der Aufbau der anderen Ethernet-Pakete unterscheidet sich nur in den Feldern „Type“ und „Data Bytes“, denen je nach Pakettyp eine andere Funktion zukommt.

Die Netzteilnehmer verarbeiten nur diejenigen Pakete weiter, die tatsächlich an sie selbst adressiert sind.

Logische Adressierung und Datentransport

Zur Erinnerung: Jede Ethernet-Adresse wird vom Hersteller weltweit einmalig in das entsprechende Endgerät eingebrannt. Damit ist jedes Endgerät im Netzwerk eindeutig adressierbar.

In einem Verbund mehrerer Einzelnetzwerke bietet die Ethernet-Adresse allein aber keinen Anhaltspunkt dazu, zu welchem Netzwerk der Teilnehmer gehört. Für eine netzwerkübergreifende Kommunikation und die dazu benötigte Adressierung reicht Ethernet allein deshalb nicht aus.

Darüber hinaus arbeitet Ethernet verbindungslos: Der Absender erhält vom Empfänger keine Bestätigung, ob ein Paket angekommen ist.

Spätestens wenn ein Ethernet-Netzwerk mit anderen Netzen verbunden werden soll, muss also mit übergeordneten Protokollen – etwa mit TCP/IP – gearbeitet werden.

Bereits in den 60er-Jahren vergab das amerikanische Militär den Auftrag, ein Protokoll zu schaffen, das unabhängig von der verwendeten Hard- und Software einen standardisierten Informationsaustausch zwischen einer beliebigen Zahl verschiedener Netzwerke möglich machen sollte. Aus dieser Vorgabe entstand im Jahr 1974 das Protokoll TCP/IP.

Obwohl TCP und IP immer in einem Wort genannt werden, handelt es sich hier um zwei aufeinander aufsetzende Protokolle. Das Internet Protocol IP übernimmt die richtige Adressierung und Zustellung der Datenpakete, während das darauf aufsetzende Transport Control Protocol TCP für den Transport und die Sicherung der Daten zuständig ist.

TCP/IP im lokalen Netz

Der besseren Übersichtlichkeit halber wollen wir zunächst den Datentransport und die logische Adressierung mit TCP/IP innerhalb eines lokalen Netzes näher beleuchten.

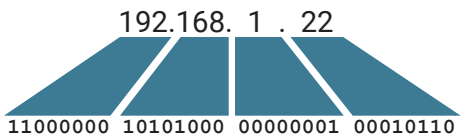
IP - Internet Protocol

Für das Verständnis der Adressierung innerhalb eines lokalen Netzes reicht uns zunächst ein Blick auf die grundsätzliche Struktur des Internet Protocols IP und auf

das Address Resolution Protocol ARP, welches die Zuordnung von IP-Adressen zu Ethernet-Adressen ermöglicht.

IP-Adressen

Unter IP hat jeder Netzwerkteilnehmer eine einmalige IP-Adresse, die oft auch als „IP-Nummer" bezeichnet wird. Diese Internet-Adresse ist ein 32-Bit-Wert, der zur besseren Lesbarkeit immer in Form von vier durch Punkte getrennte Dezimalzahlen (8-Bit-Werten) angegeben wird (Dot-Notation).

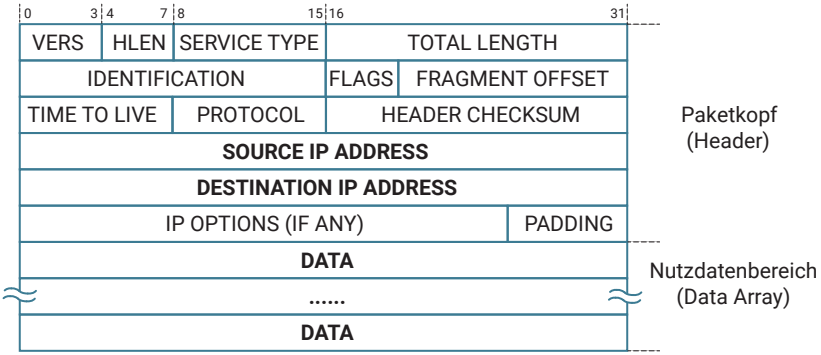


Jede IP-Adresse muss im gesamten verbundenen Netzwerk einmalig sein.

IP-Datenpakete

Auch IP-Datenpakete haben einen Rahmenaufbau und enthalten neben den zu transportierenden Nutzdaten im Paketkopf eine Fülle von Adress- und Zusatzinformationen. Wir beschränken uns hier auf die Erklärung der wichtigsten Adressinformationen.

Aufbau eines IP-Datenpakets:



Source IP address: IP-Adresse des Absenders
Destination IP address: IP-Adresse des Empfängers

ARP – Address Resolution Protocol

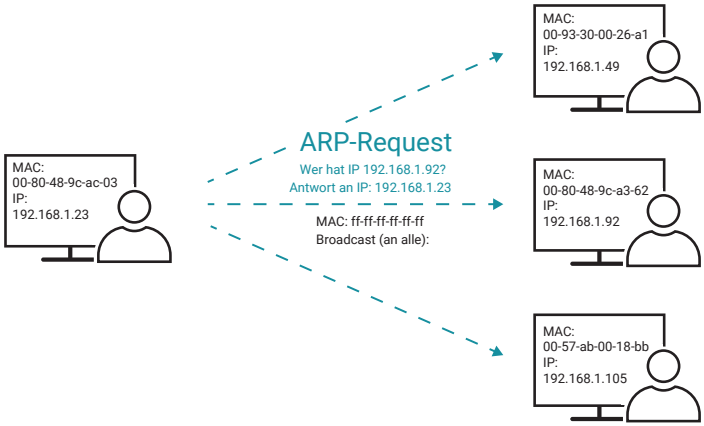
Der IP-Treiber übergibt neben dem IP-Datenpaket auch die physikalische Ethernet-Adresse an den Ethernet-Kartentreiber. Zur Ermittlung der Ethernet-Adresse des Empfängers bedient sich der IP-Treiber des Address Resolution Protocol ARP.

In jedem TCP/IP-fähigen Rechner gibt es eine ARP-Tabelle. Die ARP-Tabelle wird vom TCP/IP-Treiber bei Bedarf aktualisiert und enthält die Zuordnung von IP-Adressen zu Ethernet-Adressen.

```
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
```

Soll ein IP-Paket verschickt werden, sieht der IP-Treiber zunächst nach, ob die gewünschte IP-Adresse bereits in der ARP-Tabelle vorhanden ist. Ist dies der Fall, gibt der IP-Treiber die ermittelte Ethernet-Adresse zusammen mit seinem IP-Paket an den Ethernet-Kartentreiber weiter.

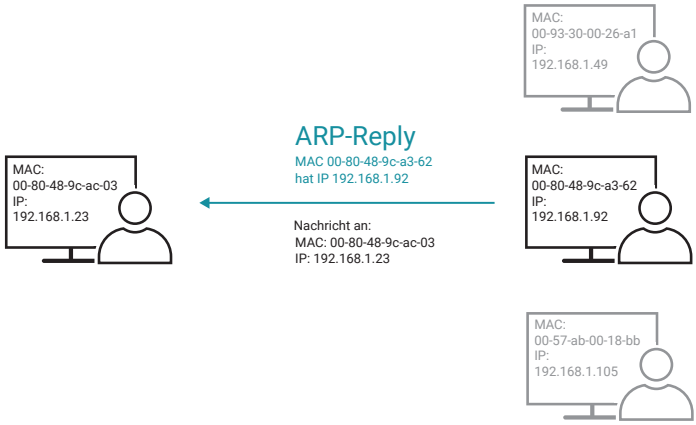
Kann die gewünschte IP-Adresse nicht gefunden werden, startet der IP-Treiber einen ARP-Request. Ein ARP-Request ist ein Rundruf (auch Broadcast genannt) an alle Teilnehmer im lokalen Netz.



Damit der Rundruf von allen Netzteilnehmern zur Kenntnis genommen wird, gibt der IP-Treiber als Ethernet-Adresse FF-FF-FF-FF-FF-FF an. Ein an FF-FF-FF-FF-FF-FF adressiertes Ethernet-Paket wird grundsätzlich von allen Netzteilnehmern gelesen. Als

Destination wird die gewünschte IP-Adresse angegeben und im Feld Protocol des Ethernet-Headers die Kennung für ARP ausgewiesen.

Derjenige Netzteilnehmer, der in diesem ARP-Request seine eigene IP-Adresse wiedererkennt, bestätigt das mit einem ARP-Reply. Der ARP-Reply ist ein auf Ethernet-Ebene an den ARP-Request-Absender adressiertes Datenpaket mit der ARP-Kennung im Protocol-Feld. Im Datenbereich des ARP-Paketes sind zusätzlich die IP-Adressen von Sender und Empfänger des ARP-Reply eingetragen.



Der IP-Treiber kann nun die dem ARP-Reply entnommene Ethernet-Adresse der gewünschten IP-Adresse zuordnen und trägt sie in die ARP-Tabelle ein.

Im Normalfall bleiben die Einträge in der ARP-Tabelle nicht dauerhaft bestehen. Wird ein eingetragener Netzwerkteilnehmer über eine bestimmte Zeit (unter Windows ca. 2 Min.) nicht kontaktiert, wird der entsprechende Eintrag gelöscht. Das hält die ARP-Tabelle schlank und ermöglicht den Austausch von Hardwarekomponenten unter Beibehaltung der IP-Adresse. Man nennt diese zeitlich begrenzten Einträge auch dynamische Einträge.

Neben den dynamischen Einträgen gibt es auch statische Einträge, die der Benutzer selbst in der ARP-Tabelle anlegt. Die statischen Einträge können genutzt werden, um an neue Netzwerkkomponenten, die noch keine IP-Adresse haben, die gewünschte IP-Adresse zu übergeben.

Diese Art der Vergabe von IP-Adressen lassen auch W&T Com-Server zu: Empfängt ein Com-Server, der noch keine eigene IP-Adresse hat, ein IP-Datenpaket, das auf

Ethernet-Ebene an ihn adressiert ist, wird die IP-Adresse dieses Pakets ausgewertet und als eigene IP-Adresse übernommen.

Achtung: Nicht alle Netzwerkkomponenten besitzen diese Fähigkeit. PCs lassen sich auf diese Weise z.B. nicht konfigurieren!

TCP - Transport Control Protocol

Die Frage, auf welche Art und Weise Daten transportiert werden sollen, lösen Transportprotokolle, die jeweils verschiedenen Anforderungen gerecht werden.

Weil IP ein ungesichertes, verbindungsloses Protokoll ist, arbeitet es oft mit dem aufgesetzten TCP zusammen. TCP übernimmt die gesicherte Zustellung der Nutzdaten. Außerdem stellt TCP für die Dauer der Datenübertragung eine Verbindung zwischen zwei Netzteilnehmern her. Beim Verbindungsaufbau werden Bedingungen wie z.B. die Größe der Datenpakete festgelegt, die für die gesamte Verbindungsdauer gelten.

TCP kann man mit einer Telefonverbindung vergleichen. Teilnehmer A wählt Teilnehmer B an; Teilnehmer B akzeptiert mit dem Abheben des Hörers die Verbindung, die dann bestehen bleibt, bis einer der beiden sie beendet.

TCP arbeitet nach dem sogenannten *Client/Server-Prinzip*:

Den Netzteilnehmer, der eine Verbindung aufbaut (der also die Initiative ergreift), bezeichnet man als Client. Der Client nimmt einen vom Server angebotenen Dienst in Anspruch, wobei je nach Dienst ein Server auch mehrere Clients gleichzeitig bedienen kann.

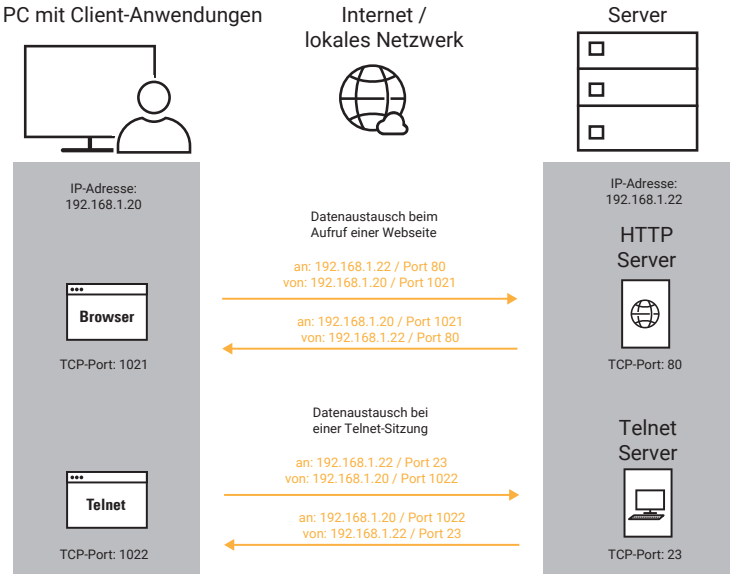
Der Netzteilnehmer, zu dem der Client die Verbindung aufbaut, wird als Server bezeichnet. Ein Server tut von sich aus nichts, sondern wartet auf einen Client, der eine Verbindung zu ihm aufbaut. Im Zusammenhang mit TCP spricht man von TCP-Client und TCP-Server.

TCP sichert die übertragenen Nutzdaten mit einer Checksumme und versieht jedes gesendete Datenpaket mit einer Sequenznummer. Der Empfänger eines TCP-Pakets prüft anhand der Checksumme den korrekten Empfang der Daten. Hat ein TCP-Server ein Paket korrekt empfangen, wird über einen vorgegebenen Algorithmus aus der Sequenznummer eine Acknowledgement-Nummer errechnet.

Die Acknowledgement-Nummer wird dem Client mit dem nächsten selbst gesende-

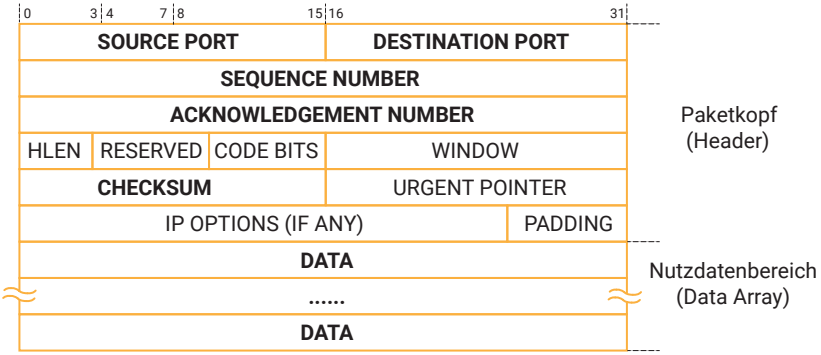
ten Paket als Quittung zurückgegeben. Der Server versieht seine gesendeten Pakete ebenfalls mit einer eigenen Sequenznummer, die wiederum vom Client mit einer Acknowledgement-Nummer quittiert wird. Dadurch ist gewährleistet, dass der Verlust von TCP-Paketen bemerkt wird, und diese im Bedarfsfall in korrekter Abfolge erneut gesendet werden können.

Darüber hinaus leitet TCP die Nutzdaten auf dem Zielrechner an das richtige Anwendungsprogramm weiter. Dazu nutzt TCP Portnummern, kurz Ports. Unterschiedliche Anwendungsprogramme – auch Dienste genannt – sind über unterschiedliche Portnummern ansprechbar. So ist Telnet z.B. über Port 23, HTTP, der Dienst, über den Webseiten aufgerufen werden, über Port 80 zu erreichen. Vergleicht man ein TCP-Paket mit einem Brief an eine Behörde, kann man die Portnummer mit der Raumnummer der adressierten Dienststelle vergleichen. Befindet sich z.B. das Straßenverkehrsamt in Raum 312 und man adressiert einen Brief an eben diesen Raum, dann gibt man damit zugleich auch an, dass man die Dienste des Straßenverkehrsamts in Anspruch nehmen möchte.



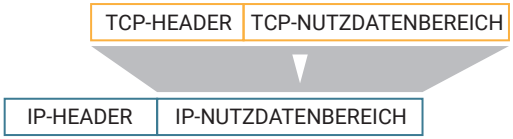
Damit die Antwort des Zielrechners wieder an der richtigen Stelle ankommt, hat auch die Client-Anwendung eine Portnummer. Bei PC-Anwendungen werden die Portnummern der Client-Anwendungen dynamisch und unabhängig von der Art der Anwendung vergeben.

Auch TCP verpackt die Nutzdaten in einen Rahmen von Zusatzinformationen. Solche TCP-Pakete sind wie folgt aufgebaut:



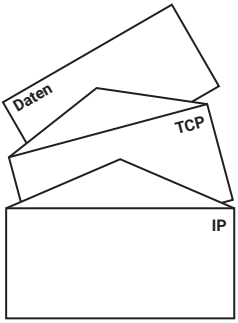
Source Port:	Portnummer der Applikation des Absenders
Destination Port:	Portnummer der Applikation des Empfängers
Sequence No:	Offset des ersten Datenbytes relativ zum Anfang des TCP-Stroms (garantiert die Einhaltung der Reihenfolge)
Acknowledgement No:	im nächsten TCP-Paket erwartete Sequence No.
Data:	Nutzdaten

Das so entstandene TCP-Paket wird in den Nutzdatenbereich eines IP-Pakets eingesetzt.



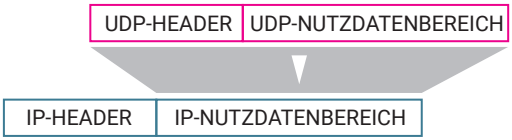
Das IP-Paket hat anschließend folgenden Aufbau:

Die Nutzdaten werden quasi in einen Briefumschlag (TCP-Paket) gesteckt, der in einen weiteren Briefumschlag (IP-Paket) gesteckt wird.



UDP – User Datagramm Protocol

UDP ist ein weiteres Transportprotokoll, das genau wie TCP auf IP aufsetzt.



Das IP-Paket hat anschließend folgenden Aufbau:

Im Gegensatz zu TCP arbeitet UDP verbindungslos. Das heißt, jedes Datenpaket wird als Einzelsendung behandelt und es gibt keine Rückmeldung darüber, ob ein Paket beim Empfänger angekommen ist. UDP-Datenpakete werden auch als Datagramm bezeichnet.

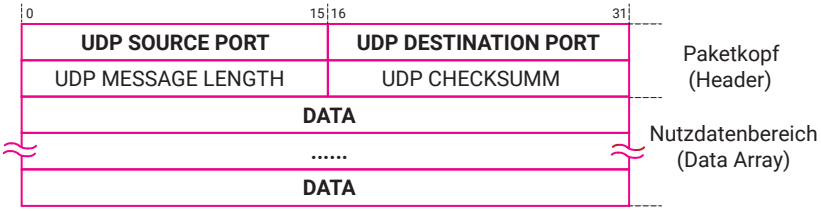
Bei UDP gibt es kein Client/Server-Prinzip. Man spricht bei beiden Kommunikationspartnern von UDP-Peers.

Weil unter UDP keine Verbindungen auf- und abgebaut werden müssen und somit keine Timeout-Situationen entstehen können, kann UDP jedoch schneller als TCP sein: Wenn ein Paket verloren geht, wird die Datenübertragung hier eben ungehindert fortgesetzt, sofern nicht ein höheres Protokoll für Wiederholungen sorgt.

Die Datensicherheit ist unter UDP also in jedem Fall durch das Anwendungspro-

gramm zu gewährleisten.

UDP-Datenpakete sind durch den Verzicht auf datensichernde Informationen deutlich kleiner als z.B. bei TCP.



Source Port: Portnummer der sendenden Anwendung (Rücksendeport für Empfänger)

Destination Port: Zielport, an den die Daten beim Empfänger übertragen werden sollen

Als Faustregel kann man sagen:

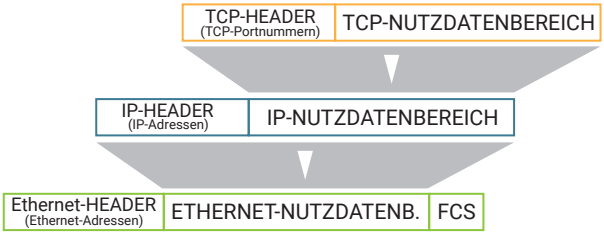
- Für kontinuierliche Datenströme oder große Datenmengen sowie in Situationen, in denen ein hohes Maß an Datensicherheit gefordert ist, wird in aller Regel TCP eingesetzt.
- Bei häufig wechselnden Übertragungspartnern sowie einer Gewährleistung der Datensicherheit durch übergeordnete Protokolle ist der Einsatz von UDP effektiver.

Der Weg eines Zeichens durch das Ethernet

Wir haben nun mit TCP/IP (bzw. UDP/IP) das Handwerkszeug kennengelernt, mit dem Daten adressiert und transportiert werden. Zusammenfassend wird im Folgenden noch einmal der Weg eines Zeichens durch ein lokales Netz aufgezeigt.

TCP/IP ist ein rein logisches Protokoll und benötigt immer eine physikalische Grundlage. Wie bereits anfänglich erwähnt, genießt Ethernet heute die größte Verbreitung bei den physikalischen Netzwerktopologien. So findet man auch in den meisten TCP/IP-Netzwerken Ethernet als physikalische Grundlage.

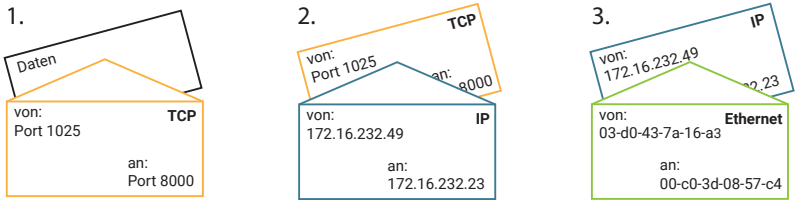
TCP/IP und Ethernet werden zusammengeführt, indem jedes TCP/IP-Paket in den Nutzdatenbereich eines Ethernet-Paketes eingebettet wird.



Das komplette Paket sieht dann so aus:

Die Nutzdaten passieren auf ihrem Weg von der Applikation auf dem PC bis ins Netzwerk mehrere Treiberebenen:

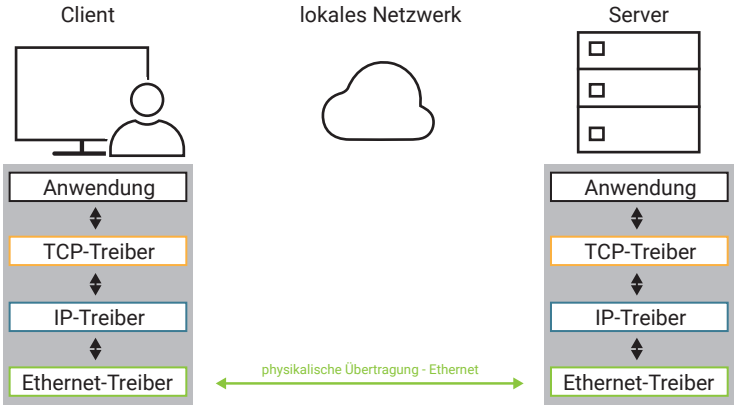
- Das Anwendungsprogramm übergibt die vom Anwender konfigurierte IP-Adresse und den zugehörigen TCP-Port dem TCP/IP-Treiber (oft auch TCP/IP-Stack genannt).
- Der TCP/IP-Treiber koordiniert den Aufbau der TCP-Verbindung.
- Die vom Anwendungsprogramm übergebenen Nutzdaten werden vom TCP-Treiber je nach Größe in kleinere übertragbare Blöcke geteilt.
- Jeder Datenblock wird zunächst vom TCP-Treiber in ein TCP-Paket verpackt *(1.)*.
- Der TCP-Treiber übergibt das TCP-Paket und die IP-Adresse des Empfängers an den IP-Treiber.
- Der IP-Treiber verpackt das TCP-Paket in ein IP-Paket *(2.)*.
- Der IP-Treiber sucht in der ARP-Tabelle (Address Resolution Protocol) nach der Ethernet-Adresse des durch die IP-Adresse angegebenen Empfängers (wenn kein Eintrag vorhanden ist, wird zunächst ein ARP-Request ausgelöst) und übergibt das IP-Paket zusammen mit der ermittelten Ethernet-Adresse an den Ethernet-Kartentreiber.
- Der Ethernet-Kartentreiber verpackt das IP-Paket in ein Ethernet-Paket und gibt dieses Paket über die Netzwerkkarte auf das Netzwerk aus *(3.)*.



Beim Empfänger findet die Prozedur in umgekehrter Reihenfolge statt:

- Die Ethernet-Karte erkennt an der Destination-Ethernet-Adresse, dass das Paket für den Netzteilnehmer bestimmt ist und gibt es an den Ethernet-Treiber weiter.
- Der Ethernet-Treiber isoliert das IP-Paket und gibt es an den IP-Treiber weiter.
- Der IP-Treiber isoliert das TCP-Paket und gibt es an den TCP-Treiber weiter.
- Der TCP-Treiber überprüft den Inhalt des TCP-Paketes auf Richtigkeit und übergibt die Daten anhand der Portnummer an die richtige Applikation.

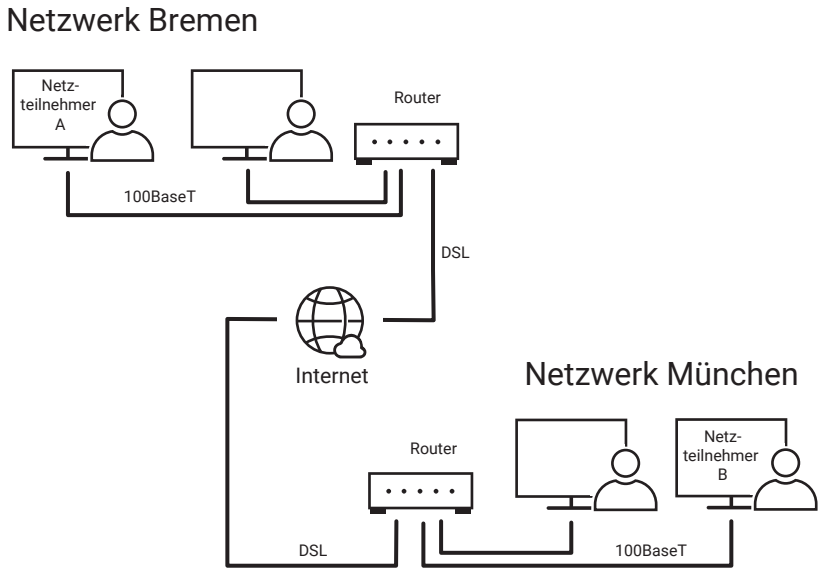
Das Beispiel zeigt das Zusammenspiel von logischer Adressierung (TCP/IP) und tatsächlicher physikalischer Adressierung (Ethernet).



Erst dieses Zusammenspiel macht es möglich, netzübergreifend und hardwareunabhängig Daten auszutauschen.

TCP/IP bei netzübergreifender Verbindung

Das Internet-Protokoll macht es möglich, eine unbestimmte Anzahl von Einzelnetzen zu einem Gesamtnetzwerk zusammenzufügen. Es ermöglicht also den Datenaustausch zwischen zwei beliebigen Netzteilnehmern, die jeweils in beliebigen Einzelnetzen positioniert sind. Die physikalische Ausführung der Netze bzw. Übertragungswege (Ethernet, Token Ring, DSL, ...) spielt hierbei keine Rolle.



Die verschiedenen Einzelnetze werden über Gateways/Router miteinander verbunden und fügen sich so zum Internet bzw. Intranet zusammen. Die Adressierung erfolgt nach wie vor über die IP-Adresse, die wir uns nun einmal genauer ansehen werden.

Netzklassen

Die IP-Adresse unterteilt sich in Net-ID und Host-ID, wobei die Net-ID zur Adressierung des Netzes und die Host-ID zur Adressierung des Netzteilnehmers innerhalb eines Netzes dient. An der Net-ID erkennt man, ob der Empfänger, zu dem die Verbindung aufgebaut werden soll, im gleichen Netzwerk wie der Sender zu finden ist. Stimmt dieser Teil der IP-Adresse bei Sender und Empfänger überein, befinden sich beide im selben Netzwerk; stimmt er nicht überein, ist der Empfänger in einem anderen Netzwerk zu finden.

Ähnlich sind auch Telefonnummern aufgebaut. Hier unterscheidet man ebenfalls zwischen Vorwahl und Teilnehmerrufnummer.

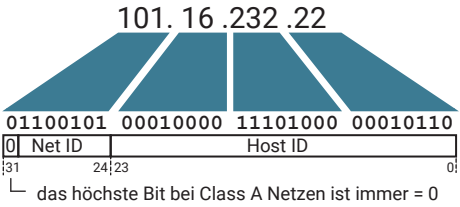
Je nachdem, wie groß der Anteil der Net-ID an einer IP-Adresse ist, sind wenige große Netze mit jeweils vielen Teilnehmern und viele kleine Netze mit jeweils wenigen Teilnehmern denkbar. In den Anfängen des Internets hat man den IP-Adressraum anhand der Größe der möglichen Netzwerke in Klassen unterschieden.

Klasse	Adresse	mögliche Netze	mögliche Hosts
Class A	1.xxx.xxx.xxx - 126.xxx.xxx.xxx	127 (2^{7})	ca. 16 Millionen (2^{24})
Class B	128.0.xxx.xxx - 191.255.0.0	ca. 16000 (2^{14})	ca. 65000 (2^{16})
Class C	192.0.0.xxx - 223.255.255.xxx	ca. 2 Millionen (2^{21})	254 (2^{8} - 2)

Zwei der möglichen Host-Adressen entfallen jeweils für die Netzwerkadresse (z.B. 192.168.1.0) und die Broadcast-Adresse (z.B. 192.168.1.255 - dazu später mehr) des Netzes.

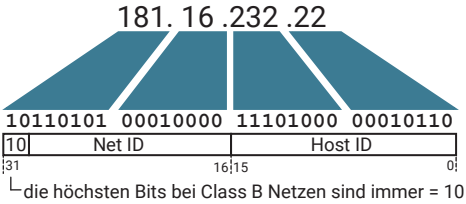
Class A

Das erste Byte der IP-Adresse dient der Adressierung des Netzes, die letzten drei Bytes adressieren den Netzteilnehmer.



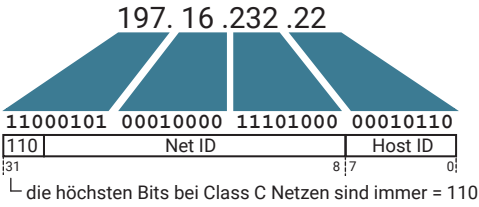
Class B

Die ersten zwei Bytes der IP-Adresse dienen der Adressierung des Netzes, die letzten zwei Bytes adressieren den Netzteilnehmer.



Class C

Die ersten drei Bytes der IP-Adresse dienen der Adressierung des Netzes, das letzte Byte adressiert den Netzteilnehmer.



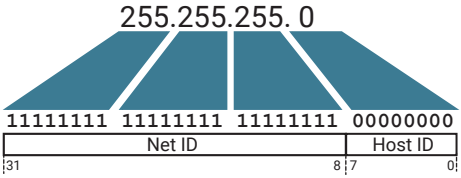
Neben den hier aufgeführten Netzen gibt es auch noch Class-D- und Class-E-Netze, deren Adressbereiche oberhalb der Class-C-Netze liegen. Class-D-Netze und Class-E-Netze haben in der Praxis wenig Bedeutung, da sie nur zu Forschungszwecken und für Sonderaufgaben verwendet werden. Der normale Internetnutzer kommt mit diesen Netzwerkklassen nicht in Berührung.

Subnet-Mask

Nun ist es allerdings möglich, ein Netzwerk – egal welcher Netzwerkklasse – in weitere Unternetzwerke zu unterteilen. Zur Adressierung solcher Subnets reicht die von den einzelnen Netzwerkklassen vorgegebene Net-ID allerdings nicht aus; man muss einen Teil der Host-ID zur Adressierung der Unternetze abzweigen. Im Klartext bedeutet dies, dass die Net-ID sich vergrößert und die Host-ID entsprechend kleiner wird.

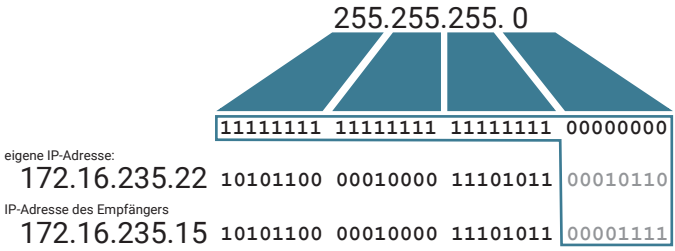
Welcher Teil der IP-Adresse als Net-ID und welcher als Host-ID ausgewertet wird, gibt die Subnet-Mask vor. Die Subnet-Mask ist genau wie die IP-Adresse ein 32-Bit-

Wert, der in Dot-Notation dargestellt wird. Betrachtet man die Subnet-Mask in binärer Schreibweise, ist der Anteil der Net-ID mit Einsen, der Anteil der Host-ID mit Nullen aufgefüllt.



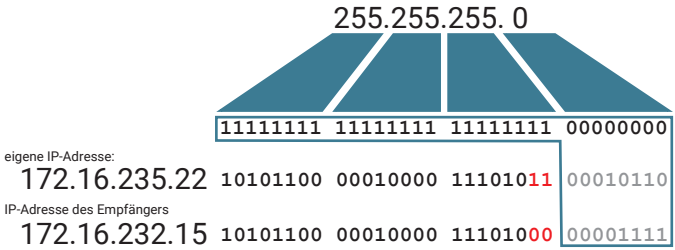
Bei jedem zu verschickenden Datenpaket vergleicht der IP-Treiber die eigene IP-Adresse mit der des Empfängers. Hierbei werden die Bits der Host-ID über den mit Nullen aufgefüllten Teil der Subnet-Mask ausgeblendet.

Sind die ausgewerteten Bits beider IP-Adressen identisch, befindet sich der gewählte Netzteilnehmer im selben Subnet.



Im oben dargestellten Beispiel kann der IP-Treiber die Ethernet-Adresse über ARP ermitteln und diese dem Netzwerkkarten-Treiber zur direkten Adressierung übergeben.

Unterscheidet sich auch nur ein einziges der ausgewerteten Bits, befindet sich der gewählte Netzteilnehmer nicht im selben Subnet.



In diesem Fall muss das IP-Paket zur weiteren Vermittlung ins Zielnetzwerk einem Gateway bzw. Router übergeben werden. Zu diesem Zweck ermittelt der IP-Treiber über ARP die Ethernet-Adresse des Routers, auch wenn im IP-Paket selbst nach wie vor die IP-Adresse des gewünschten Netzteilnehmers eingetragen ist.

Gateways und Router

Gateways bzw. Router sind im Prinzip nichts anderes als Computer mit zwei Netzwerkkarten. Ethernet-Datenpakete, die auf Karte A empfangen werden, werden vom Ethernet-Treiber entpackt und das enthaltene IP-Paket wird an den IP-Treiber weitergegeben. Dieser prüft, ob die Ziel-IP-Adresse zum an Karte B angeschlossenen Subnet gehört und das Paket direkt zugestellt werden kann, oder ob das IP-Paket an ein weiteres Gateway übergeben wird.

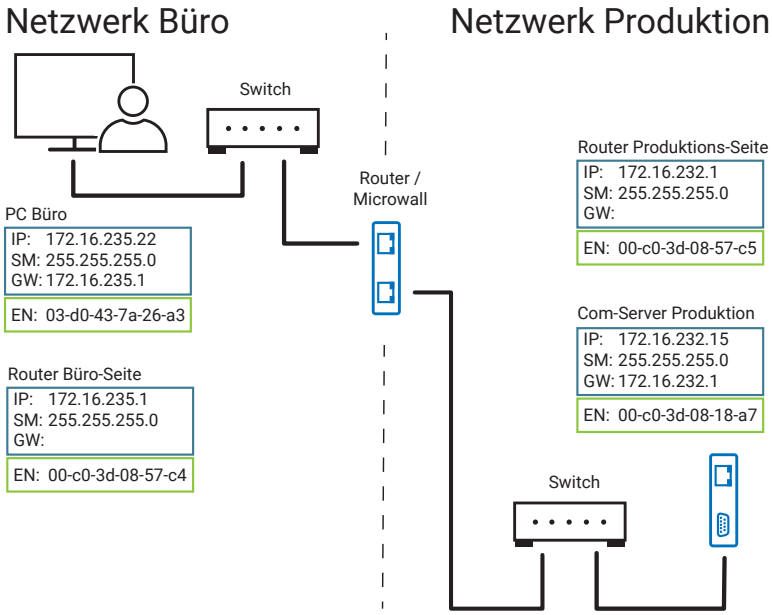
So kann ein Datenpaket auf seinem Weg von einem Netzteilnehmer zum anderen mehrere Gateways/Router passieren. Während auf IP-Ebene auf der gesamten Strecke die IP-Adresse des Empfängers eingetragen ist, wird auf Ethernet-Ebene immer nur das nächste Gateway adressiert. Erst auf dem Teilstück vom letzten Gateway/Router zum Empfänger wird in das Ethernet-Paket die Ethernet-Adresse des Empfängers eingesetzt.

Neben Routern, die ein Ethernet-Subnet mit einem anderen Ethernet-Subnet verbinden, gibt es auch Router, die das physikalische Medium wechseln – z.B. von Ethernet auf DSL. Während auch hier die IP-Adressierung über die gesamte Strecke gleich bleibt, ist die physikalische Adressierung von einem Router zum anderen den auf den Teilstrecken geforderten physikalischen Gegebenheiten angepasst.

Zwischen zwei DSL-Routern arbeitet die Infrastruktur des entsprechenden Internet-Providers. Die physikalische Adressierung erfolgt dann zum Beispiel über Anschlusskennungen, welche die eindeutige Zuordnung des jeweiligen DSL-Anschlusses sicherstellen.

Routing - Der Weg der Daten durch mehrere Netze

Im folgenden Abschnitt wird anhand einer bestehenden Telnet-Verbindung der Weg eines Zeichens über eine geroutete Netzwerkverbindung beschrieben.



Wir gehen in unserem Beispiel davon aus, dass ein Anwender im Büro-Netzwerk bereits eine Telnet-Verbindung zu einem W&T Com-Server im Produktionsnetzwerk aufgebaut hat; die Verbindung zwischen den Netzwerken wird über einen entsprechend konfigurierten Router hergestellt.

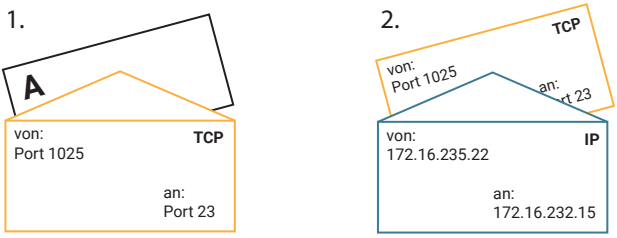
Der Anwender am PC gibt in der Telnet-Client-Anwendung das Zeichen „A“ ein.

- Das Telnet-Client-Programm auf dem PC übergibt dem TCP/IP-Stack das „A“ als Nutzdatenteil. Die IP-Adresse des Empfängers (172.16.232.15) und die Portnummer 23 für Telnet wurden dem TCP/IP-Stack bereits bei Aufbau der Verbindung übergeben.

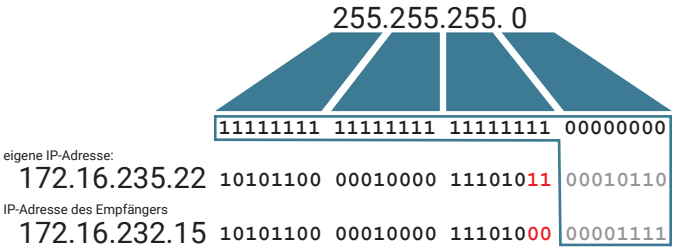
- Der TCP-Treiber schreibt das „A“ in den Nutzdatenbereich eines TCP-Pakets und trägt als Destination-Port die 23 ein *(1.)*.

- Der TCP-Treiber übergibt das TCP-Paket und die IP-Adresse des Empfängers an den IP-Treiber.

- Der IP-Treiber verpackt das TCP-Paket in ein IP-Paket *(2.)*.



- Der IP-Treiber ermittelt über den Vergleich der Net-ID-Anteile von eigener IP-Adresse und IP-Adresse des Empfängers, ob das IP-Paket im eigenen Subnet zugestellt werden kann oder einem Router übergeben wird.



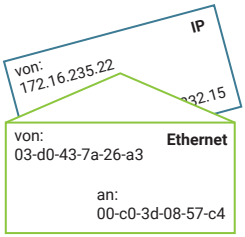
  Hier sind die Net-ID-Anteile der beiden Adressen nicht gleich; das IP-Paket muss folglich an den eingetragenen Router übergeben werden.

- Der IP-Treiber ermittelt über ARP die Ethernet-Adresse des Routers. Da die TCP-Verbindung bereits aufgebaut ist, wird die IP-Adresse des Routers bereits in der ARP-Tabelle aufgelöst sein.

```
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
```

- Der IP-Treiber entnimmt der ARP-Tabelle die Ethernet-Adresse des Routers und übergibt sie zusammen mit dem IP-Paket dem Ethernet-Kartentreiber.

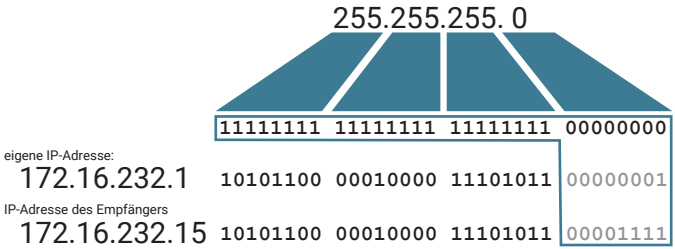
- Der Ethernet-Kartentreiber verpackt das IP-Paket in ein Ethernet-Paket und gibt dieses Paket über die Netzwerkkarte auf das Netzwerk aus.



- Der Router entnimmt dem empfangenen Ethernet-Paket das IP-Paket.

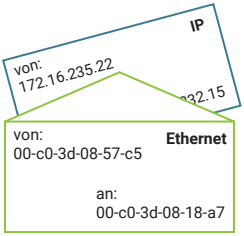
- Die IP-Adresse des Empfängers wird mit einer sogenannten Routing-Tabelle verglichen. Anhand dieser Routing-Tabelle entscheidet der Router, ob er das IP-Paket Richtung Zielnetz weitervermitteln kann. Je nach Netzwerkinfrastruktur kann es sein, das ein IP-Paket mehrere Router durchläuft, bis es im Zielnetzwerk ankommt.

- Für den Ethernet-Anschluss Richtung Zielnetzwerk stellt der Router über IP-Adressen und Subnet-Mask fest, ob das empfangene IP-Paket auf einem der lokalen Ethernet-Ports ins Zielnetzwerk zugestellt werden kann oder einem weiteren Router übergeben werden muss.



In unserem Beispiel hat das IP-Paket das Zielnetzwerk erreicht und kann am entsprechenden Ethernet-Port ausgegeben und über Ethernet adressiert werden.

- Der Router, der intern zum Zielnetzwerk hin ebenfalls eine ARP-Tabelle führt, ermittelt über ARP die zur IP-Adresse passende Ethernet-Adresse und verpackt das im Adressierungsbereich immer noch unveränderte IP-Paket in ein Ethernet-Paket.



- Der Com-Server erkennt an der Destination-Ethernet-Adresse, dass das Paket für ihn bestimmt ist, und entnimmt das IP-Paket.

- Der IP-Treiber des Com-Servers isoliert das TCP-Paket und gibt es an den TCP-Treiber weiter.

- Der TCP-Treiber überprüft den Inhalt des TCP-Paketes auf Richtigkeit und übergibt die Daten – in diesem Fall das „A“ – an den seriellen Treiber des Com-Servers.

- Der serielle Treiber gibt das „A“ auf der seriellen Schnittstelle aus.

Bei einer TCP-Verbindung wird der korrekte Empfang eines Datenpaketes mit dem Rücksenden einer Acknowledgement-Nummer quittiert. Das Quittungspaket durchläuft den gesamten Übertragungsweg und alle damit verbundenen Prozeduren in Gegenrichtung. All dies spielt sich innerhalb weniger Millisekunden ab.

VLAN - Virtual Local Area Network

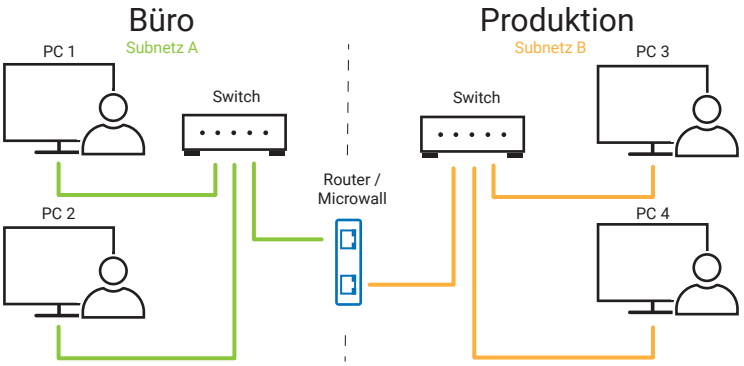
Wenn größere Netzwerke in kleinere Subnetze unterteilt werden, ist das meist auch mit einer räumlichen Aufteilung der Subnetze verbunden.

So können z.B. in einer Firma der Bürobereich und die Produktion jeweils eigene Subnetze A und B bekommen, wobei die Subnetze über einen Router miteinander verbunden werden.

PCs und andere Endgeräte, die im Bürobereich an einen Switch angesteckt werden, sind somit ausschließlich mit Subnetz A verbunden. Im Produktionsbereich angeschlossene Endgeräte gehören zu Subnetz B.

Neben der logischen Trennung gibt es so auch eine physische Trennung, die darin besteht, dass alle an einem Switch angeschlossenen Endgeräte zum selben Subnetz gehören.

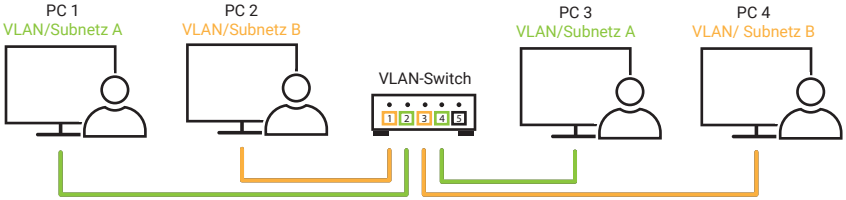
Soll z.B. ein PC in der Büroabteilung an das Subnetz der Produktion angebunden werden, müsste dazu ein Netzwerkkabel von einem Switch, der zum Produktions-Subnetz gehört, zu dem PC im Büro verlegt werden.



Durch die Nutzung von VLANs wird diese starre physische Trennung der Subnetze aufgehoben. Switches mit VLAN-Unterstützung erlauben es, den Netzwerkverkehr verschiedener Subnetze innerhalb des Switches auf bestimmte Ausgangsports zu verteilen. Dazu wird jeder Anschluss des Switches einem ausgewählten Subnetz zugeordnet. So werden virtuelle Subnetze oder auch VLANs gebildet.

Portbasierte VLANs

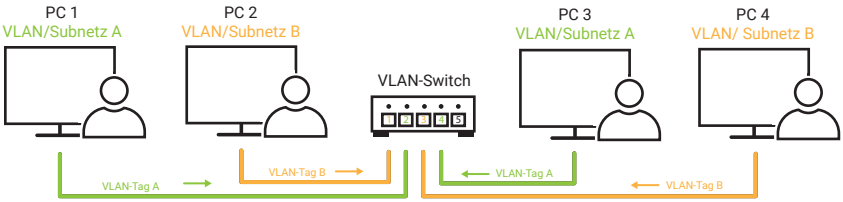
Beim portbasierten VLAN wird durch entsprechende Konfiguration jeder Port eines Switches einem bestimmten Subnetz bzw. VLAN zugewiesen. Netzwerkteilnehmer, die an diesem Port angeschlossen sind, können nur auf das so konfigurierte VLAN/Subnetz zugreifen.



Tagged VLANs

Während beim portbasierten VLAN die Zuordnung Port/VLAN im Switch fest gespeichert ist, bestimmt beim tagged VLAN der angeschlossene Netzteilnehmer, mit welchem VLAN er verbunden wird. Das geschieht über ein Tag, das im Header des

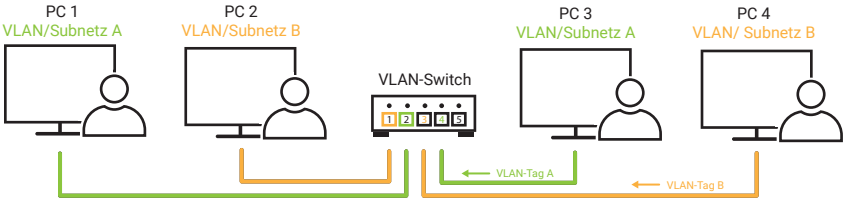
Ethernet-Datenpaketes mitgesendet wird.



Tagged VLAN setzt voraus, dass die Netzwerk-Hardware und der Netzwerktreiber des Endgerätes und der verwendete Switch tagged VLAN unterstützen.

Portbasiertes und tagged VLAN im Mischbetrieb

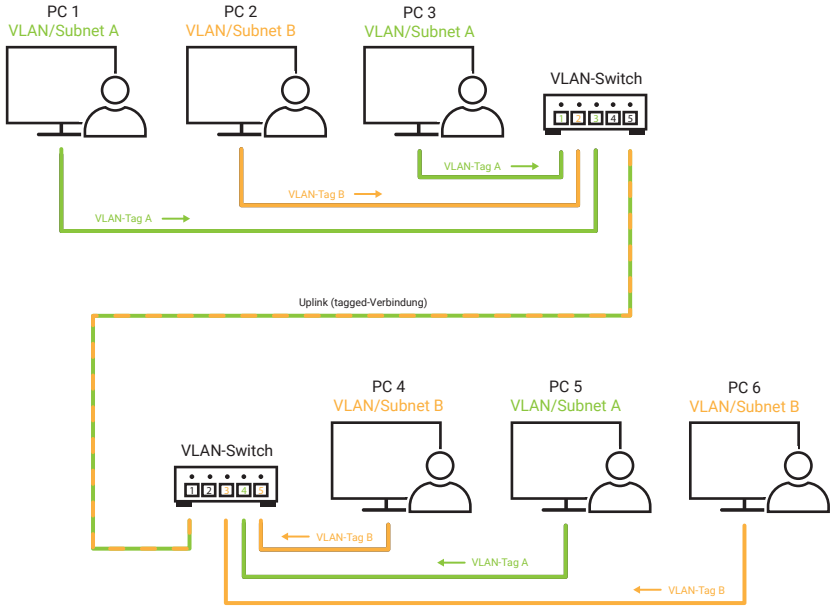
Die meisten VLAN-fähigen Switches erlauben beide Varianten parallel, so dass Geräte, die kein tagged VLAN unterstützen, portbasiert angeschlossen werden können, während andere Geräte über das Tag bestimmen, zu welchem VLAN sie gehören.



VLANs über mehrere Switches

Die Ausdehnung von VLANs kann auch über mehr als einen Switch hinaus gehen. Beim rein portbasierten VLAN wird allerdings für jedes VLAN ein separates Ethernet-Kabel zwischen den Switches benötigt, wobei der zugehörige Port am Switch entsprechend für das VLAN konfiguriert sein muss.

In der Praxis spart man sich diesen doppelten Verkabelungsaufwand und nutzt stattdessen das tagged Verfahren, um Switche im VLAN-Umfeld miteinander zu verbinden. Die dazu verwendeten Ports an den Switches müssen entsprechend konfiguriert sein.

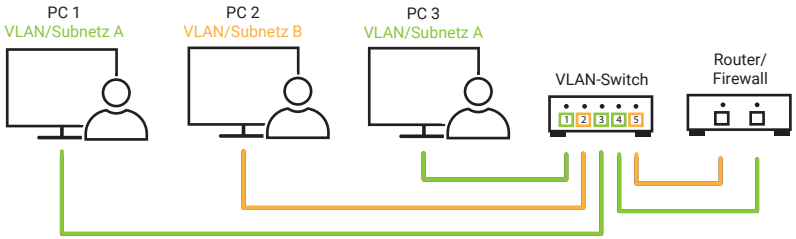


Für die tagged Verbindung zwischen den Switches spielt es keine Rolle, ob die einzelnen Netzteilnehmer tagged oder portbasiert angebunden sind.

Zwischen zwei Switches kann auch mehr als eine tagged Verbindung eingesetzt werden, um einen besseren Datendurchsatz zu erreichen. Dieses Verfahren bezeichnet man als Trunking.

VLANs und Routing

Sollen die Nutzer von zwei VLANs untereinander Daten austauschen, geht das nur mittels Routing. Dazu kann ein Standard-Router mit entsprechend konfigurierten Ports des Switches verbunden werden.



Alternativ gibt es besondere Switches, die das Routing zwischen den VLANs intern selbst abwickeln.

Schutz durch Firewalls

Die Grundfunktion eines Routers beschränkt sich auf das Vermitteln von IP-Datenpaketen von einem Netzwerk zum anderen. Das erfolgt wie gezeigt durch den Abgleich von Empfangs-IP-Adresse, Net-ID und Subnet-Mask. Sind die Voraussetzungen für das Routing erfüllt, werden die Datenpakete ungefiltert weitergegeben.

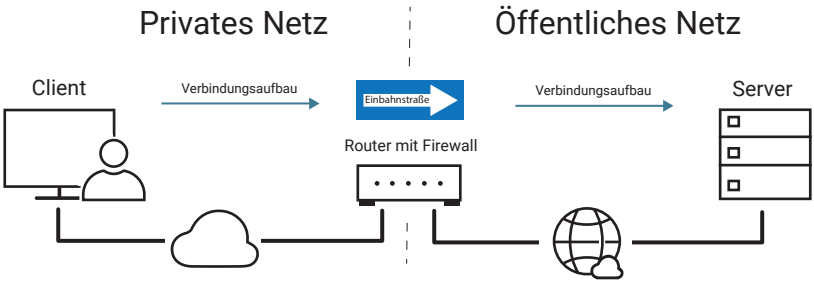
Bei Netzwerken, die über einen Router mit dem Internet verbunden sind, hätte jeder von überall aus Zugriff in das lokale Netzwerk. Das wäre natürlich fatal für die Datensicherheit.

Deshalb übernehmen die meisten Router gleichzeitig die Funktion einer Firewall. Für die Firewall können Regeln konfiguriert werden, die bestimmen, welche Datenpakete in welche Richtung weitervermittelt werden.

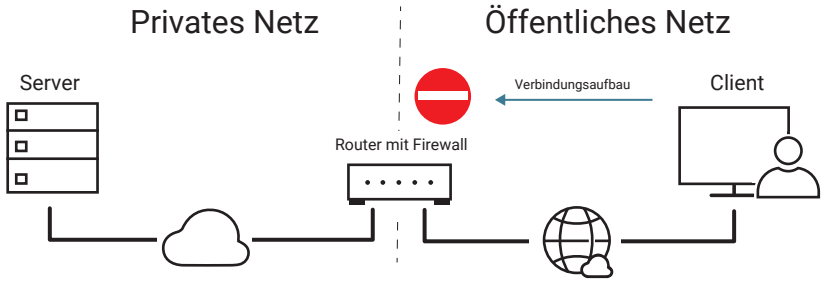
Die Konfigurationsmöglichkeiten von Firewalls sind so vielfältig, dass sie als Stoff für ein eigenes Buch reichen würden. Deshalb beschränken wir uns hier auf das Wesentlichste.

Die am häufigsten verwendeten Firewalls sind inzwischen Router, die ein lokales Netzwerk über die Anschlusstechnik des Providers direkt mit dem Internet verbinden (z.B. DSL-Router).

Von Hause aus sind solche Router so konfiguriert, dass Verbindungen aus dem lokalen Netz ins Internet uneingeschränkt möglich sind (Client im lokalen Netz, Server im Internet).



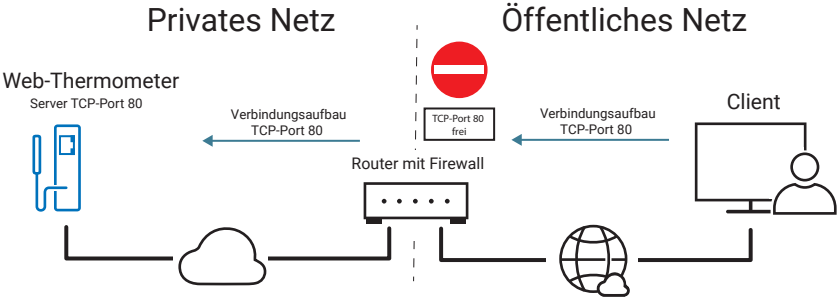
Verbindungen aus dem Internet ins lokale Netzwerk hingegen werden von der Firewall abgewiesen (Client im Internet, Server im lokalen Netz).



Es gibt aber auch Fälle, bei denen bestimmte Server-Dienste im lokalen Netz für Clients aus dem öffentlichen Netz erreichbar sein sollen.

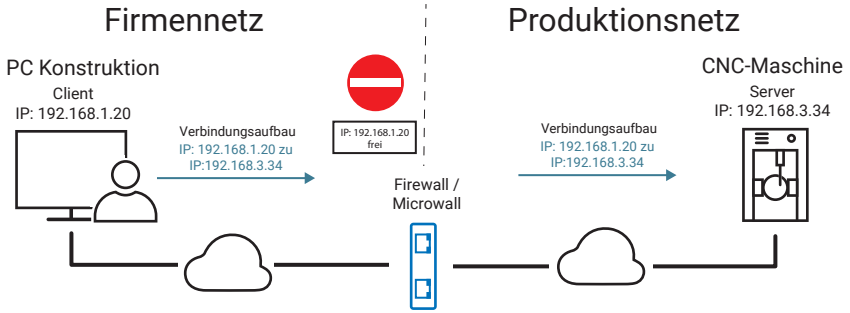
Soll zum Beispiel die Webseite eines W&T Web-Thermometers für den Zugriff aus dem Internet freigegeben werden, kann eine entsprechende Regel konfiguriert werden:

„Auf Port 80 (HTTP zur Browser-Kommunikation) darf von außen eine TCP-Verbindung aufgebaut werden."



Insbesondere in großen Firmennetzwerken werden Firewalls aber auch eingesetzt, um Teilnetzwerke abzuschotten, zum Beispiel um in der Produktion netzwerkgesteuerte CNC-Maschinen vor Fremdzugriff zu schützen. Die Konfigurationsmöglichkeiten für solche Firewalls gehen meist deutlich weiter als das reine Sperren und Öffnen bestimmter Ports.

So kann zum Beispiel der Zugriff aus dem allgemeinen Netzwerk in das Teilnetzwerk auf einen bestimmten Netzteilnehmer beschränkt werden.



Die Regeln lassen sich noch feiner abstimmen. So können Freigaben für Kombinationen aus IP-Adresse und Port erfolgen. Auch IP-Adressbereiche können definiert werden. Natürlich ist eine Vielzahl von Regeln für beide Verbindungsrichtungen möglich.

Zu beachten ist, dass die Regeln für den TCP-Verbindungsaufbau gelten.

Ist die TCP-Verbindung einmal hergestellt, können Daten in beide Richtungen ausgetauscht werden.

Selbstverständlich können auch Regeln für UDP-Datagramme aufgestellt werden. Je nach Firewall gibt es dazu verschiedene Möglichkeiten.

Noch einmal zur Erinnerung: Bei UDP gibt es keine Verbindung und damit nicht automatisch einen definierten Rückkanal. Deshalb müssen für die UDP-Kommunikation oft für beide Richtungen Regeln konfiguriert werden, damit der Teilnehmer auf der anderen Seite der Firewall auf ein UDP-Datagramm antworten kann.

Einige Firewalls erkennen aber auch Absende-IP-Adresse und Absende-Port und geben den Rückkanal auch bei UDP automatisch frei.

NAT - Network Address Translation

Möchte man über einen normalen Router ein Netzwerk mit zehn Endgeräten z.B. mittels DSL mit dem Internet verbinden, so würde jedes dieser Endgeräte eine eigene, einmalige IP-Adresse benötigen.

Wie bereits angesprochen, sind öffentliche IP-Adressen, d.h. solche, die von der IANA einmalig vergeben werden und daher mit dem Internet verbunden werden können, inzwischen knapp.

Neben diesen öffentlichen IP-Adressen gibt es jedoch noch einen Adressraum für private Netze. Die Bezeichnung „privat“ steht hier für „nicht öffentlich“ und schließt auch Firmennetze mit ein. Je nach Netzwerkgröße sind für private Netze diese Adressbereiche vorgesehen:

für Class A: Netze 10.0.0.1 bis 10.255.255.254
für Class B: Netze 172.16.0.1 bis 172.31.255.254
für Class C: Netze 192.168.0.1 bis 192.255.255.254

In diesen Adressbereichen können sich Administratoren bei der Einrichtung ihres privaten Netzwerks frei bedienen. Da ein und dieselbe Adresse in mehreren Netzwerken vorkommen kann, sind Adressen aus diesen Bereichen nur innerhalb des eigenen Netzwerkes eindeutig. Somit ist auch kein normales Routing zu diesen Adressen möglich. Genau hier schafft NAT-Routing Abhilfe.

Mit NAT (Network Address Translation) wurde eine Art des Routings geschaffen, die es erlaubt, eine Vielzahl von Teilnehmern in einem privaten Netzwerk zum Internet hin mit nur einer öffentlichen IP-Adresse zu repräsentieren.

Zur Erinnerung: Bei normalem TCP/IP-Datenverkehr adressiert die IP-Adresse den Netzwerkteilnehmer; die Portnummer die Anwendung im Gerät.

Beim NAT-Routing wird auch die Portnummer als zusätzliche Adressinformation für das Endgerät selbst mitgenutzt.

Client im privaten Netzwerk
Die Arbeitsweise von NAT-Routing soll hier anhand eines kleinen Beispiels erläutert werden.

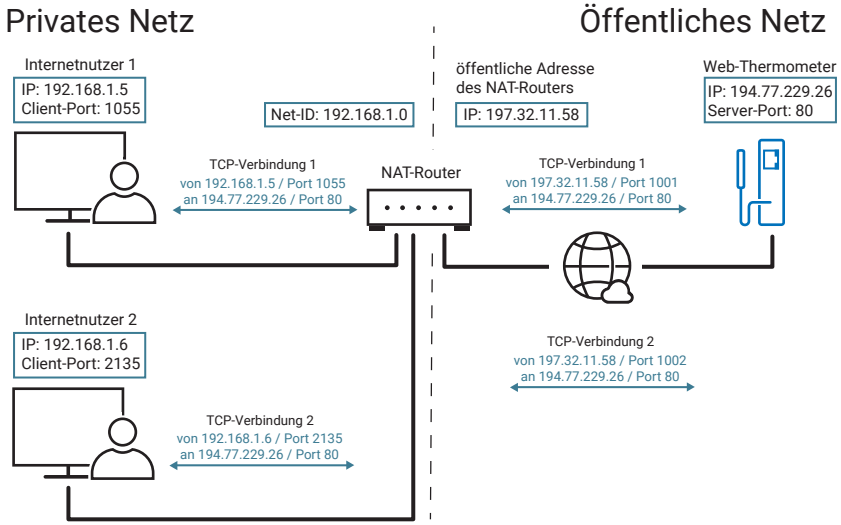
In einem privaten Class C Netzwerk wird im Adressraum 192.168.1.x gearbeitet. Als Übergang zum Internet ist ein NAT-Router im Einsatz, der nach außen mit der IP-Adresse 197.32.11.58 arbeitet.

Der PC mit der netzinternen IP-Adresse 192.168.1.5 baut eine TCP-Verbindung zum W&T Web-Thermometer (IP 194.77.229.26, Port 80) im Internet auf und benutzt dazu den lokalen Port 1055.

Ein zweiter PC mit der IP-Adresse 192.168.1.6 baut ebenfalls eine TCP-Verbindung zum Web-Thermometer auf und benutzt dazu den lokalen Port 2135.

Um mit dem Web-Thermometer verbunden zu werden, wenden sich die PCs zunächst an den NAT-Router.

Der NAT-Router wechselt in den TCP/IP-Datenpaketen, die zum Web-Thermometer weitergesendet werden, die IP-Adresse des jeweiligen PCs gegen seine eigene öffentliche IP-Adresse aus. Auch die vom PC vorgegebene Port-Nr. kann gegen eine vom NAT Router verwaltete Port-Nr. ausgetauscht werden.



Die vergebenen Portnummern verwaltet der NAT-Router in einer Tabelle, die folgendermaßen aufgebaut ist:

```
nach aussen   im privaten Netz
Port-Nr.      zugehörige IP    zugehörige Port-Nr.
1001          192.168.1.5      1055
1002          192.168.1.6      2135
```

Das Web-Thermometer empfängt also für beide Verbindungen Datenpakete, in denen der NAT-Router als Absender eingetragen ist. Dabei wird aber für jede Verbindung jeweils eine eigene Port-Nr. verwendet.

In alle Datenpakete in Richtung der beiden PCs setzt das Web-Thermometer diese

„verbogenen“ Adressinformationen ein. Das bedeutet, die TCP/IP-Pakete werden so aufgebaut, dass der NAT-Router der Empfänger ist.

Empfängt der NAT-Router ein solches an ihn adressiertes Datenpaket, stellt er mit Hilfe der Zuordnungstabelle fest, wer der tatsächliche Empfänger ist, und ersetzt die empfangenen Adressdaten durch die ursprünglichen netzinternen Verbindungsparameter.

Die Zuordnungstabelle für ausgehende Verbindungen (Client im privaten Netz, Server außerhalb) wird dynamisch verwaltet und kann natürlich viel mehr als zwei Verbindungen beinhalten. So können beliebig viele Verbindungen nach außen geroutet werden.

Server im privaten Netzwerk

Datenverkehr in die andere Richtung (Server im privaten Netz, Client außerhalb) kann natürlich genauso über NAT abgewickelt werden.

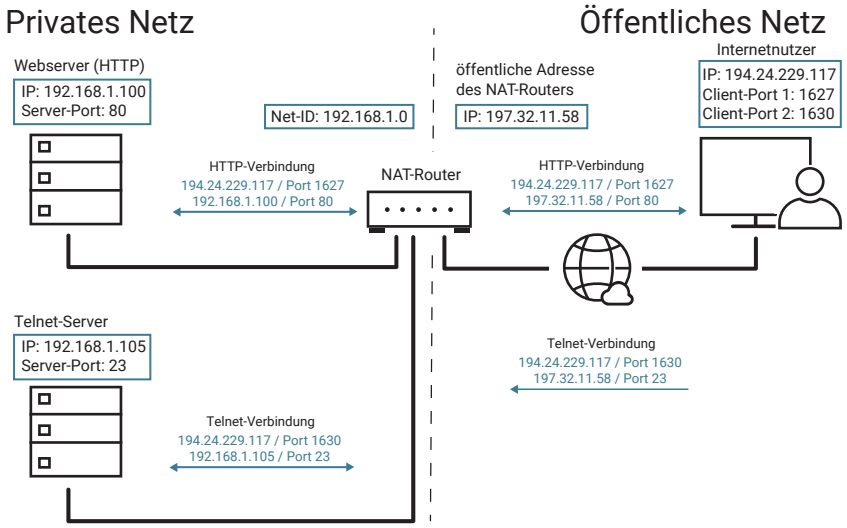
Auch hier wird mit Hilfe einer Zuordnungstabelle bestimmt, zu welchem Endgerät und auf welchen Port eingehende Verbindungsanforderungen bzw. Datenpakete geroutet werden sollen.

Im Gegensatz zu der Zuordnungsliste für ausgehende Verbindungen ist die Server-Zuordnungsliste statisch. Damit Server im privaten Netz von außen erreichbar sind, muss der Administrator einen entsprechenden Eintrag in der Liste anlegen.

Für jeden Server-Dienst, der aus dem öffentlichen Netz zugänglich sein soll, ist ein Eintrag in der Serverliste nötig.

Sollen von außen z.B. ein Webserver (HTTP = Port 80) und ein Telnet-Server (Telnet = Port 23) erreichbar sein, könnte die Server-Tabelle so aussehen:

```
nach aussen   im privaten Netz
Server Port   zugehörige IP      zugehörige Port-Nr
80            192.168.1.100      80
23            192.168.1.105      23
```



Detaillierte Informationen zu den Protokollen Telnet und HTTP folgen in den Kapiteln Anwendungsprotokolle und Web-Protokolle.

Den Austausch der Verbindungsparameter nimmt der NAT-Router genauso vor, wie bei den im vorhergehenden Abschnitt gezeigten Verbindungen.

In einem privaten Netzwerk, das über einen NAT-Router mit nur einer IP-Adresse zum Internet abgebildet wird, darf natürlich jeder Server-Port nur einmal in der Server-Tabelle vorkommen. Das bedeutet, dass ein spezieller Server-Dienst mit einer spezifischen Portnummer nur von einem internen Endgerät angeboten werden kann.

Port-Forwarding

Port-Forwarding kann als eine erweiterte Form des NAT-Routing betrachtet werden. Während beim NAT Routing der nach außen repräsentierte Port im privaten Netzwerk beibehalten wird, ändert Port-Forwarding auch die Portnummer.

Beispiel: Im privaten Netzwerk sind zwei Server mit den IP-Adressen 192.168.1.100 und 192.168.1.105 in Betrieb. Beide Server sind innerhalb des privaten Netzwerks über Port 80 als HTTP-Server erreichbar.

Zusätzlich sollen beide Server auch aus dem Internet erreichbar sein. Das geht natürlich nicht über den gleichen Port. Der Router muss also mindestens einen der Ser-

ver nach außen über einen abweichenden Port repräsentieren.

```
nach aussen   im privaten Netz
Server Port   zugehörige IP      zugehörige Port-Nr
80            192.168.1.100      80
81            192.168.1.105      80
```

Port-Forwarding wird allerdings nicht von allen Routern unterstützt.

Übertragungsprotokolle

In den vorangegangenen Kapiteln haben wir uns mit Ethernet als Übertragungsprotokoll, IP als Protokoll zur logischen Adressierung und TCP/UDP als aufgesetzte Protokolle zum Datentransport bzw. zur Transportsicherung beschäftigt.

Außerdem wurde netzwerkübergreifender Datenaustausch und das damit verbundene Routing beschrieben.

In Zeiten des Internets ist Routing aber fast immer auch damit verbunden, das Teilstück zwischen zwei Netzwerken mit einer anderen Technik als Ethernet zu überbrücken. Zum Beispiel mit DSL, über das Mobilfunknetz oder andere physikalische Standards.

Die dazu genutzte Physik wollen wir hier erst mal außen vor lassen (mehr dazu im Kapitel „Der Weg ins Internet“).

Unabhängig vom physikalischen Standard ist bei der Datenfernübertragung (DFÜ) aber immer ein übergeordnetes Protokoll nötig, welches folgende Aufgaben übernimmt:

- Aufbau einer logischen Verbindung zwischen beiden Standorten
- Authentifizierung (Prüfen der Zugangsberechtigung)
- Aufbereitung des eingehenden Datenverkehrs für die Übertragung und Wiederherstellen des ursprünglichen Datenformates am Ende der Übertragungsstrecke
- Datensicherung
- ggf. Verschlüsselung der Übertragungsdaten
- Abbau der logischen Verbindung nach Abschluss der Datenübertragung

SLIP - Serial Line IP Protocol

Ein erster Ansatz für die Übertragung von DFÜ-Daten war SLIP. SLIP ist ein sehr einfaches Protokoll, das ausschließlich für den Transport von IP-Datenverkehr geeignet ist und nicht alle oben aufgeführten Anforderungen erfüllt.

Die kompletten IP-Datenpakete werden bei SLIP einfach um ein festgelegtes Start- und Endezeichen erweitert. Zufällig im IP-Paket vorkommende Zeichen dieses Typs ersetzt der Sender durch eine Kombination aus Ersatzzeichen.

So präpariert wird Paket für Paket auf die Leitung gegeben.

An den Start-/Endezeichen eines Pakets erkennt der Empfänger, wo das eigentliche IP-Paket beginnt bzw. endet. Die Ersatzzeichen werden vom Empfänger wieder gegen das Original ausgetauscht und die Start-/Endezeichen entfernt.

Durch die Beschränkung auf IP-Datenübertragung und fehlende Sicherheitsmechanismen wird SLIP heute für normale Internetzugänge nicht mehr benutzt. Dort wo räumlich abgesetzte Netzwerksegmente über Distanzen verbunden werden sollen, die mit einer normalen Ethernet-Verkabelung nicht mehr möglich sind, kann SLIP aber nach wie vor eine zweckmäßige und preiswerte Lösung sein.

Die W&T Com-Server können z.B. als SLIP-Router konfiguriert werden und somit TCP/IP-Daten über eine RS232- oder RS422-Verkabelung übertragen.

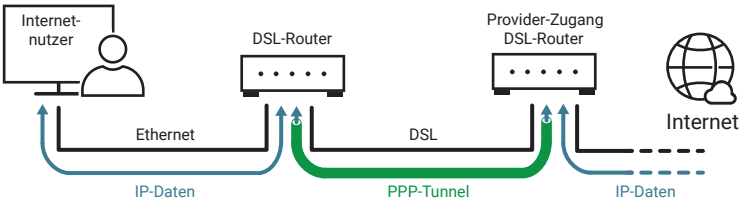
PPP - Point-to-Point Protocol

Eine wesentliche Eigenschaft von PPP ist, dass neben IP-Daten auch Daten in Form anderer Protokolle wie z.B. IPX/SPX usw. übertragen werden können.

Das bedeutet, Daten beliebigen Formats müssen unverändert zwischen zwei Netzwerkstandorten ausgetauscht werden. Diese Technik bezeichnet man auch als Datentunnelung.

Sowohl für den Zugang ins Internet als auch zur Verbindung mit einem entfernten, nicht öffentlichen Netzwerk sorgt PPP für die dazu benötigte gesicherte Datenübertragung.

PPP stellt sozusagen einen Tunnel durch die netzwerkfremde Umgebung her.



Protokollablauf

Der Aufbau einer PPP-Verbindung findet in mehreren Schritten statt und benötigt eine bestehende physikalische Verbindung wie z.B. DSL:

1. Aushandeln der Verbindungsoptionen
   Um festzulegen, mit welchen Optionen PPP arbeiten soll, wird das LCP-Protokoll (Link Control Protocol) benutzt.
   Verhandelbar sind unter anderem:
   - Art der Authentifizierung
   - Blockgröße der Übertragungsdaten
   - Datenkompression
   - Art der zu übertragenden Daten (IP, IPX, ...)
2. Authentifizierung
   Hierbei werden User-ID und ein Passwort übergeben. Es gibt zwei Arten der Passwortübergabe:
   - PAP - Password Authentication Protocol
     Passwortübergabe lesbar im Klartext
   - CHAP - Challenge Handshake
     verschlüsselte Passwortübergabe
3. Konfiguration übergeordneter Netzwerkprotokolle
   Soll über PPP eine Verbindung in Netze mit übergeordneten Protokollen (z.B. Internet-Protokoll) hergestellt werden, ist es erforderlich, bestimmte das entsprechende Protokoll betreffende Einstellungen vorzunehmen.
   Die nötigen Informationen werden mittels des NCP (Network Control Protocol) übergeben. Im Falle eines Internetzugangs über PPP wird als NCP das internetspezifische IPCP (Internet Protocol Control Protocol) verwendet. IPCP erlaubt z.B. für die Dauer der PPP-Verbindung die Vergabe einer IP-Adresse.
4. Übertragung der Nutzdaten
   Sobald alle Verbindungsoptionen festgelegt sind und der Nutzer seine Zugangsberechtigung nachgewiesen hat, beginnt der eigentliche Austausch von Nutzdaten.
   Im Fall einer Verbindung zum Internet können das Daten in Form aller IP-basierenden Protokolle sein (UDP, TCP, Telnet, FTP, HTTP...).
5. Abbau der PPP-Verbindung
   Auch der Verbindungsabbau wird über LCP abgewickelt.

Protokollaufbau

Ähnlich wie Ethernet bettet PPP die zu transportierenden Daten in eine festgelegte Paketstruktur:

Flag (1 Byte)
Startzeichen zur Paketsynchronisation bzw. Paketerkennung

Address (1 Byte)
Eine Punkt-zu-Punkt-Verbindung erfordert keine Adressierung.

Trotzdem ist dieses Feld aus Kompatibilitätsgründen zu anderen Netzwerkprotokollen vorhanden, wird aber von PPP nicht benutzt und ist willkürlich mit dem Wert 255 gefüllt.

Control (1 Byte)
Dieses Feld war ursprünglich zur Nummerierung der Pakete vorgesehen, hat bei PPP aber immer den Wert 3, da ohne Paketnummerierung gearbeitet wird.

Protocol (1 oder 2 Bytes)
Der Inhalt dieses Feldes gibt an, wie das aktuelle PPP-Paket genutzt wird: Verbindungsaufbau, Steuerinformation, Authentifizierung, Datentransport, Verbindungsabbau, ...

Information (n Bytes)
An dieser Stelle wird die eigentliche Information (z.B. IP-Daten) übertragen. Bei Steuerpaketen stehen hier die Steueroptionen im LCP-Format.

Die Größe dieses Feldes ist per LCP verhandelbar, beträgt in aller Regel aber 1500 Bytes. Ist die zu transportierende Information kleiner, wird mit Füllzeichen aufgefüllt.

FCS (2 Bytes)
Checksumme zur Kontrolle der empfangenen Daten

Flag (1 Byte)
Endezeichen zur Paketsynchronisation

Durch die Möglichkeit, über eine PPP-Verbindung verschiedene unabhängige IP-Dienste und Protokolle gleichzeitig zu übertragen, können auch ganze Netzwerke über PPP miteinander verbunden werden. Dazu werden allerdings geeignete Router benötigt.

Hilfsprotokolle

Nachdem im vorangegangenen Kapitel die grundlegenden Protokolle der TCP/IP-Datenübertragung erklärt wurden, soll im Folgenden auf die Anwendungsprotokolle eingegangen werden, die auf diese Basisprotokolle aufsetzen.

Bei den Anwendungsprotokollen unterscheidet man zwischen Hilfsprotokollen und tatsächlichen Anwendungsprotokollen.

Hilfsprotokolle werden für Management- und Diagnosezwecke genutzt und laufen oft für den Anwender unsichtbar im Hintergrund ab.

Zu den Hilfsprotokollen zählen:

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

DHCP - Dynamic Host Configuration Protocol

Zur Erinnerung: Jedes Ethernet-Endgerät hat eine weltweit einmalige Ethernet-Adresse (MAC-Adresse), die vom Hersteller unveränderbar vorgegeben wird. Für den Einsatz in TCP/IP-Netzen vergibt der Netzwerkadministrator dem Endgerät zusätzlich eine zum Netzwerk passende IP-Adresse.

Wird kein DHCP benutzt, werden die IP-Adressen „klassisch“ vergeben:

- Bei Geräten, die direkte User-Eingaben erlauben (z.B. PCs), kann die IP-Nummer direkt in ein entsprechendes Konfigurationsmenü eingegeben werden.
- Bei „Black-Box-Geräten“ (z.B. Com-Servern) gibt es zum einen das ARP-Verfahren über das Netzwerk, zum anderen besteht die Möglichkeit, die Konfigurationsinformation über eine serielle Schnittstelle einzugeben. Darüber hinaus stellen einige Hersteller Tools (z.B. das WuTility-Tool von W&T) zur Verfügung, um Embedded Geräte direkt vom PC aus zu konfigurieren.

Neben der IP-Adresse müssen als weitere Parameter noch Subnet-Mask und Gateway sowie ggf. ein DNS-Server (mehr dazu im nächsten Kapitel) konfiguriert werden.

Bei großen Netzen mit vielen unterschiedlichen Endgeräten bringt das allerdings schnell ein hohes Maß an Konfigurations- und Verwaltungsaufwand mit sich.

Mit DHCP wird dem Netzwerkadministrator ein Werkzeug angeboten, mit dem die Netzwerkeinstellungen der einzelnen Endgeräte automatisch, einheitlich und zentral konfigurierbar sind.

Für die Nutzung von DHCP wird im Netzwerk mindestens ein DHCP-Server benötigt, der die Konfigurationsdaten für einen vorgegebenen IP-Adressbereich verwaltet.

DHCP-fähige Endgeräte erfragen beim Booten von diesem Server ihre IP-Adresse und die zugehörigen Parameter wie Subnet-Mask und Gateway. DHCP-Server sehen drei grundsätzliche Möglichkeiten der IP-Adresszuteilung und Konfiguration vor:

- Vergabe der Adressen aus einem IP-Adresspool
- Vergabe einer reservierten IP-Adresse
- Ausschluss bestimmter IP-Adressen

Vergabe der IP-Adresse aus einem Adresspool

Auf dem DHCP-Server wird ein Bereich von IP-Adressen festgelegt, aus dem einem anfragenden Netzteilnehmer eine zur Zeit nicht benutzte Adresse zugeteilt wird. Die Zuteilung ist bei diesem Verfahren in aller Regel zeitlich begrenzt, wobei die Nutzungsdauer (Lease-Time) vom Netzwerkadministrator festgelegt oder ganz deaktiviert werden kann. Darüber hinaus lassen sich wichtige Daten (Lease-Time, Subnet-Mask, Gateway, DNS-Server usw.) in einem Konfigurationsprofil hinterlegen, das für alle Endgeräte gilt, die aus dem Adresspool bedient werden.

Vorteile

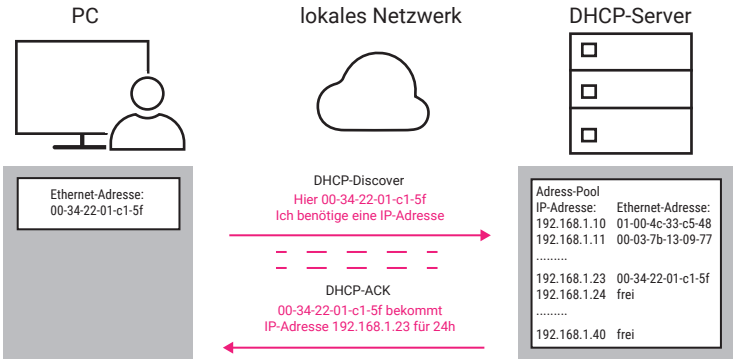
- Geringer Administrationsaufwand
- Anwender können mit demselben Endgerät ohne Konfigurationsaufwand an verschiedenen Standorten ins Netzwerk.
- Sofern nicht alle Endgeräte gleichzeitig im Netzwerk aktiv sind, kann die Anzahl der möglichen Endgeräte größer sein als die Zahl der verfügbaren IP-Adressen.

Nachteile

- Ein Netzteilnehmer kann nicht anhand seiner IP-Adresse identifiziert werden, da nicht vorhersehbar ist, welche IP-Adresse ein Endgerät beim Start zugewiesen bekommt.

Beispiel: Typische Fälle für die Vergabe von IP-Adressen aus einem Adresspool sind Universitätsnetzwerke. Hier gibt es Netze mit einer fast unbegrenzten Zahl potentieller Anwender, von denen aber nur jeweils wenige tatsächlich im Netzwerk arbeiten. Dank DHCP haben die Studierenden die Möglichkeit, ihr Notebook oder Tablet ohne Konfigurationsänderung von einem Labor ins andere mitzunehmen und im Netzwerk zu betreiben.

Um den Administrations- bzw. Konfigurationsaufwand gering zu halten, arbeiten aber auch die meisten Heimnetzwerke (ein DSL-Router, wenige PCs, Drucker und Smartphones) mit DHCP. Die Aufgabe des DHCP-Servers übernimmt hier der DSL-Router.



Vergabe einer reservierten IP-Adresse

Der Netzwerkadministrator hat die Möglichkeit, einzelne IP-Adressen für bestimmte Endgeräte zu reservieren. Auf dem DHCP-Server wird dazu der IP-Adresse die Ethernet-Adresse des Endgeräts zugeordnet; für jede reservierte IP-Adresse kann außerdem ein individuelles Konfigurationsprofil angelegt werden.

Vorteile:

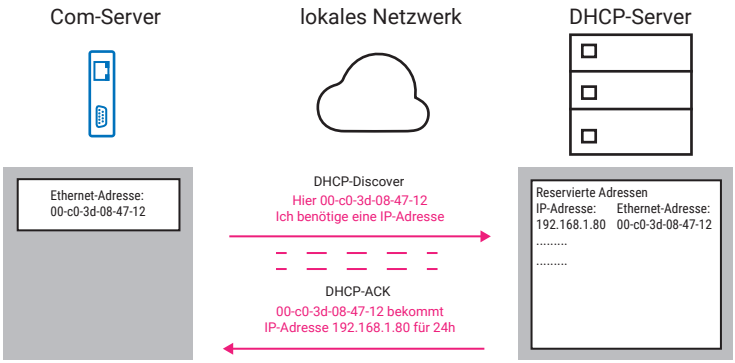
- Trotz individueller Konfiguration lassen sich alle Netzwerkeinstellungen an zentraler Stelle erledigen und müssen nicht am Endgerät selbst vorgenommen werden.
- Endgeräte können gezielt über ihre IP-Adresse angesprochen werden.

Nachteile:

- Da für jedes Endgerät spezifische Einstellungen angegeben werden müssen, steigt der Administrationsaufwand.
- Beim Austausch von Endgeräten muss auf dem DHCP-Server im Konfigurationsprofil mindestens die Ethernet-Adresse neu eingetragen werden.

Beispiel: Konfiguration von DHCP-fähigen Endgeräten wie Printservern oder Com-Servern, bei denen je nach Einsatzfall eine Adressierung über IP-Adresse benötigt wird. Im DHCP-Manager wird bei der reservierten IP-Adresse die Ethernet-Adresse des zugehörigen Endgerätes eingetragen. Beim Com-Server können als zusätzliche Parameter Subnet-Mask, Gateway (Router) und DNS-Server angegeben werden.

Hierzu muss ergänzend gesagt werden, dass einige Endgeräte auch das ältere BootP-Protokoll nutzen, um ihre Konfiguration zu erfragen. BootP ist ein Vorläufer von DHCP und wird ebenfalls von DHCP-Servern unterstützt.



Bei älteren „Black-Box-Geräten“ kann das BootP-Protokoll eingesetzt werden, um in jedem Fall die Übergabe einer reservierten IP-Adresse zu erzwingen. Ist beim DHCP-Server kein zur Ethernet-Adresse des Com-Server passender Eintrag vorhanden, sollte die BootP-Anfrage ignoriert werden und das Gerät behält die aktuell eingestellte IP-Adresse.

Leider handhaben das nicht alle DHCP-Server so und vergeben auch auf einen BootP-Request hin eine IP-Adresse aus dem Adress-Pool.

Ausschluss bestimmter IP-Adressen

Für Endgeräte, die weder DHCP- noch BootP-fähig sind, hat der Netzwerkadministrator die Möglichkeit, einzelne IP-Adressen oder auch ganze Adressbereiche von der Vergabe durch DHCP auszuschließen.

Die Konfiguration muss in diesem Fall entweder am Endgerät selbst vorgenommen werden oder durch den Einsatz mitgelieferter Tools erfolgen.

Nachteil:

- uneinheitliche und ggf. dezentrale Konfiguration
- höherer Administrationsaufwand erforderlich

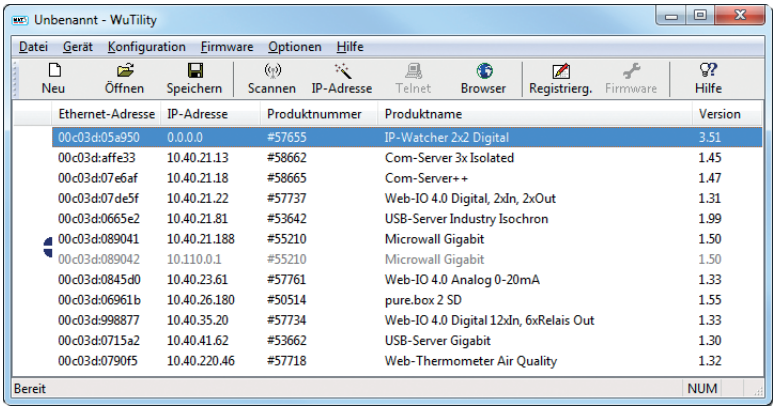
Beispiel: PCs mit älteren DOS-Versionen oder ältere Printserver sind nicht DHCP-fähig und müssen auf jeden Fall „von Hand“ konfiguriert werden.

Alle drei Verfahren können in Netzwerken mit DHCP-Unterstützung nebeneinander angewandt werden.

Natürlich gibt es auch Sonderfälle, in denen es sinnvoll ist, auf DHCP zur Adressvergabe zu verzichten. In technischen Anwendungen gilt es oft, neben der Vergabe der IP-Adressdaten noch weitere gerätespezifische Einstellungen vorzunehmen, die ohnehin nicht von DHCP unterstützt werden.

Hier bieten die vom Hersteller mitgelieferten Software-Werkzeuge in vielen Fällen mehr Komfort als DHCP.

W&T bietet dem Anwender zum Beispiel mit dem Wutility Tool ein Werkzeug zur einfachen Inbetriebnahme, Inventarisierung, Wartung und Verwaltung von W&T-Geräten wie Com-Servern, USB-Servern, Web-IO Boxen, sowie Motherboxen und pure.boxen.



Natürlich können auch solche W&T Endgeräte, die ihre IP-Adresse über DHCP bekommen haben, mit Wutility verwaltet werden.

DHCP und Router

Der Informationsaustausch zwischen Endgeräten und DHCP-Servern erfolgt auf physikalischer Ebene in Form von UDP-Broadcasts (Rundrufen ins Netz). Erstreckt sich die DHCP-Konfiguration über mehrere Subnetze, gibt es zwei Möglichkeiten:

- Der eingesetzte Router sollte als DHCP-Relay-Agent arbeiten, also das subnetz-übergreifende Weiterleiten von DHCP-Requests unterstützen.
- Es sollte in jedem Subnetz ein eigener DHCP-Server arbeiten.

DNS – das Domain Name System

Das Domain Name System ist das Adressbuch des Internets. Obwohl es vom Anwender nur im Hintergrund genutzt wird, ist es doch einer der wichtigsten Internetdienste.

Auf IP-Ebene werden die Millionen von Teilnehmern im Internet über IP-Adressen angesprochen. Für den Nutzer wäre der Umgang mit IP-Adressen aber schwierig: Wer kann sich schon merken, dass das Web-Thermometer von W&T unter der IP-Adresse 194.77.229.26 zu erreichen ist? Einen aussagekräftigen Namen, wie klima.wut.de, kann man sich dagegen viel leichter merken.

Schon in den Anfängen des Internets trug man dem Bedürfnis Rechnung, IP-Adressen symbolische Namen zuzuordnen: Auf jedem lokalen Rechner wurde eine

Hosts-Tabelle gepflegt, in der die entsprechenden Zuordnungen hinterlegt waren. Der Nachteil bestand jedoch darin, dass eben nur diejenigen Netzwerkteilnehmer erreichbar waren, deren Namen in der lokalen Liste standen. Zudem nahmen diese lokalen Listen mit dem rapiden Wachstum des Internets bald eine nicht mehr handhabbare Größe an. Man stand also vor der Notwendigkeit, ein einheitliches System zur Namensauflösung zu schaffen. Aus diesem Grund wurde 1984 der DNS-Standard verabschiedet, an dem sich bis heute kaum etwas geändert hat.

Das Prinzip ist einfach: Die Zuordnung von IP-Adressen und Domainnamen wird auf sogenannten DNS-Servern hinterlegt und dort bei Bedarf „angefragt". Doch ehe wir hier in die Details gehen, noch einige Anmerkungen zum Aufbau von Domain-Namen:

Domainnamen

Das DNS sieht eine einheitliche Namensvergabe vor, bei der jeder einzelne Host (Teilnehmer im Netz) Teil mindestens einer übergeordneten „Top-Level-Domain" ist.

Als Top-Level-Domain bietet sich ein länderspezifischer Domainname an:

- .de für Deutschland
- .at für Österreich
- .ch für Schweiz usw.

Die Domain kann aber auch nach Inhalt bzw. Betreiber gewählt werden:

- .com für kommerzielle Angebote
- .net für Netzbetreiber
- .edu für Bildungseinrichtungen
- .gov ist der US-Regierung vorbehalten
- .mil ist dem US-Militär vorbehalten
- .org für Organisationen

Alle untergeordneten (Sub-Level-) Domainnamen können vom Betreiber selbst gewählt werden, müssen in der übergeordneten Domain aber einmalig sein. Für jede Top-Level-Domain gibt es eine selbstverwaltende Institution, bei der die Sub-Level-Domains beantragt werden müssen und die damit eine Mehrfachvergabe ausschließt. Für die de-Domain ist in solchen Fragen die DENIC (Deutsches Network Information Center; http://www.denic.de) zuständig.

Ein Beispiel: klima.wut.de setzt sich zusammen aus:

- de für Deutschland als Top-Level-Domain
- wut für Wiesemann und Theis als Sub-Level-Domain
- klima für das Web-Thermometer in der Domain wut.de

Der gesamte Domainname darf maximal 255 Zeichen lang sein, wobei jeder Subdomainname höchstens 63 Zeichen umfassen darf. Die einzelnen Subdomainnamen werden mit Punkten getrennt. Eine Unterscheidung zwischen Groß- und Kleinschreibung gibt es nicht. WWW.WUT.DE führt Sie genauso auf die Homepage von W&T wie www.wut.de oder www.WuT.de.

Namensauflösung im DNS

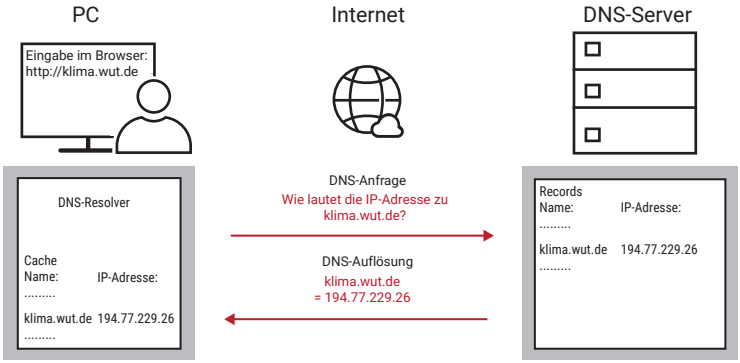
Wie bereits angesprochen, werden auf DNS-Servern (auch Nameserver genannt) Listen mit der Zuordnung von Domain-Namen und IP-Adresse geführt. Gäbe es bei den heutigen Ausdehnungen des Internets nur einen einzigen DNS-Server, wäre dieser mit der immensen Zahl der DNS-Anfragen hoffnungslos überfordert. Aus diesem Grund wird das Internet in Zonen aufgeteilt, für die ein bzw. mehrere DNS-Server zuständig sind.

Netzteilnehmer, die das DNS nutzen möchten, müssen in ihrem TCP/IP-Stack die IP-Adresse eines in ihrer Zone liegenden DNS-Servers angeben. Um auch bei Ausfall dieses Servers arbeiten zu können, verlangen die üblichen TCP/IP-Stacks sogar die Angabe eines zweiten DNS-Servers.

Welcher DNS-Server für den jeweiligen Netzteilnehmer zuständig ist, erfährt man beim Provider bzw. beim Netzwerkadministrator.

Um Domainnamen in IP-Adressen auflösen zu können, verfügen heutige TCP/IP-Stacks über ein Resolver-Programm. Gibt der Anwender anstatt einer IP-Adresse einen Domainnamen an, startet das Resolver-Programm eine Anfrage beim eingetragenen DNS-Server. Liegt dort kein Eintrag für den gesuchten Domainnamen vor, wird die Anfrage an den in der Hierarchie nächsthöheren DNS-Server weitergegeben. Dies geschieht so lange, bis die Anfrage entweder aufgelöst ist oder festgestellt wird, dass es den angefragten Domainnamen nicht gibt.

Die zum Domainnamen gehörende IP-Adresse wird von DNS-Server zu DNS-Server zurückgereicht und schließlich wieder dem Resolver–Programm übergeben. Der TCP/IP-Stack kann nun die Adressierung des Zielteilnehmers ganz normal über dessen IP-Adresse vornehmen.



Die Zuordnung von IP-Adresse und Domainnamen wird vom TCP/IP-Stack in einem Cache hinterlegt. Diese Cache-Einträge sind dynamisch: Wird der hinterlegte Netzteilnehmer für eine bestimmte Zeit nicht angesprochen, löscht der Stack den Eintrag wieder. Das hält den Cache schlank und macht es möglich, die zu einem Domainnamen gehörende IP-Adresse bei Bedarf auszutauschen.

DNS in Embedded Systemen

Nicht alle Embedded Systeme bieten die Möglichkeit, am Gerät selbst einen Domainnamen einzugeben.

Das ist auch gar nicht nötig, denn das Endgerät muss seinen eigenen Namen gar nicht wissen. Vielmehr wird die Zuordnung von Name und IP-Adresse auch hier auf dem DNS-Server festgehalten. Soll z.B. von einem Client eine Verbindung zu einem als Server arbeitendes Embedded System aufgebaut werden, erfragt der Client die zum Namen gehörende IP-Adresse wie gehabt beim DNS-Server.

Da Embedded Systeme aber häufiger in „Maschine-Maschine-Verbindungen“ als in „Mensch-Maschine-Verbindungen“ arbeiten, ist eine direkte Adressierung über IP-Adresse hier effizienter, da die Zeit für die DNS-Auflösung entfällt.

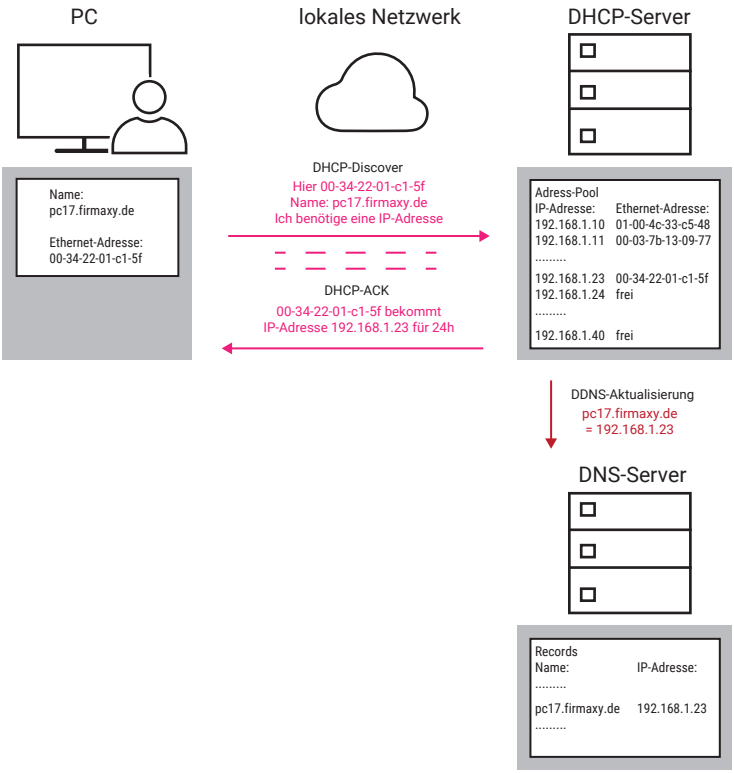
Die Adressierung über Namen ist bei Embedded Systemen nur dann sinnvoll, wenn entweder nur der Name bekannt ist (z.B. E-Mail-Adressen) oder mit einem „Umzug“ eines Servers (Name bleibt, IP-Adresse ändert sich) gerechnet werden muss (z.B. Webserver).

DDNS - dynamisches DNS in Verbindung mit DHCP

Zusammengefasst kann man sagen: DNS ist eine Art Telefonbuch fürs Netzwerk. Nun hat DNS in seiner Urform die gleichen Nachteile wie ein Telefonbuch. Ändert sich die Telefonnummer eines Teilnehmers, nachdem das Buch gedruckt wurde, kann der Teilnehmer mit Hilfe dieses nun veralteten Telefonbuches nicht mehr erreicht werden.

Die Zuordnungen in DNS-Servern werden natürlich regelmäßig aktualisiert und nicht nur einmal pro Jahr erneuert. Wird aber mit dynamischen IP-Adressen gearbeitet, die mittels DHCP vergeben werden, macht DNS nur Sinn, wenn eine ständige Korrektur der DNS-Listen betrieben wird.

Die Technik des automatischen Abgleiches zwischen DHCP-Server und DNS-Server wird als DDNS - dynamisches DNS bezeichnet. DDNS ist kein Standard-TCP/IP-Dienst.



Auf welchem Weg und in welcher Form die Synchronisation zwischen DHCP-Server und DNS-Server erfolgt, hängt davon ab, unter welchem Betriebssystem die Server laufen.

Der prinzipielle DDNS-Ablauf bei Vergabe einer IP-Adresse via DHCP verläuft folgendermaßen:

1. Das Endgerät versucht vom DHCP-Server eine IP-Adresse zu beziehen. Dabei ist der Host-Name des Gerätes (hier pc17.firmaxy.de) im Endgerät fest konfiguriert.
2. Der DHCP-Server vergibt eine IP-Adresse aus seinem Adress-Pool an das Endgerät und trägt die Zuordnung zur Ethernet-Adresse in die Adressverwaltung ein.
3. Zusätzlich übergibt der DHCP-Server dem DNS-Server IP-Adresse und Host-Namen des Endgerätes.
4. Der DNS-Server aktualisiert die Namensverwaltung mit dem neuen Eintrag.

Bei dem gezeigten Ablauf spielt es keine Rolle, ob DNS-Server und DHCP-Server auf zwei getrennten Rechnern oder auf einer gemeinsamen Hardware laufen.

Da die DDNS-Kopplung vom Netzwerkadministrator eingerichtet werden muss, kommt DDNS nur in abgeschlossenen Netzen wie z.B. Firmen-Netzen zum Einsatz.

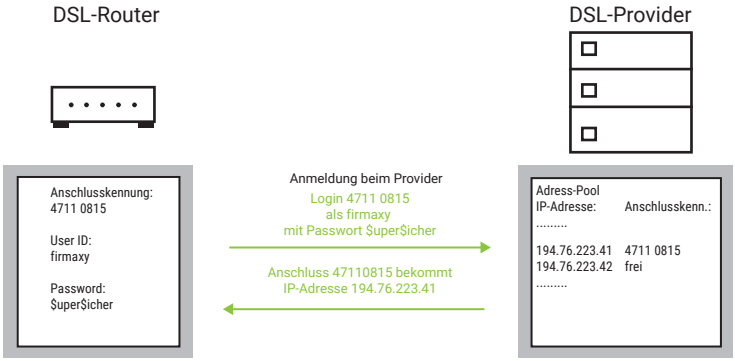
Dynamisches DNS im Internet

Nicht nur in lokalen Netzen, in denen der DHCP-Server die IP-Adressvergabe abwickelt, wird mit dynamischen IP-Adressen gearbeitet.

Zur Erinnerung: In Netzen, die miteinander verbunden sind - man spricht auch von WAN (Wide Area Network) - muss jedes angeschlossene Endgerät eine einmalige IP-Adresse haben. Diese Regel gilt insbesondere für das Internet, welches den mit Abstand größten Netzwerkverbund darstellt.

In den meisten Fällen erfolgt die Anbindung an das Internet heute mittels eines entsprechenden Routers. Der Router verbindet das lokale Ethernet-Netzwerk mit dem bereitgestellten Anschluss des Providers. Zur eindeutigen Identifikation im physischen Netz des Providers vergibt dieser für jeden Kundenanschluss eine Anschlusskennung.

Beim Einschalten des Routers teilt der Provider dem verbundenen Endgerät für die Dauer der Nutzung, ähnlich wie bei DHCP, eine IP-Adresse zu. Diese IP-Adresse wird voraussichtlich bei jeder Internetnutzung eine andere sein.



Da die meisten Internetnutzer nur Server-Dienste (E-Mail, Abruf von Webseiten, ...) in Anspruch nehmen, also Verbindungen zu diesen Servern aufnehmen, ist das kein Problem.

Soll aber das Endgerät des Internetnutzers (meist ein PC) auch für andere Internetnutzer erreichbar sein, ist die dynamische IP-Adresse ein Problem, da die aktuell zugeteilte IP-Adresse ja nur dem Provider und dem dort angekoppelten Endgerät bekannt ist.

Um dieses Problem zu umgehen, gibt es zwei Möglichkeiten:

1. Permanenter Anschluss an das Internet
Feste Internetzugänge mit einer festen IP-Adresse sind ungleich teurer als z.B. normale DSL-Zugänge. Diese Lösung bietet sich deshalb nur für größere Firmen an.

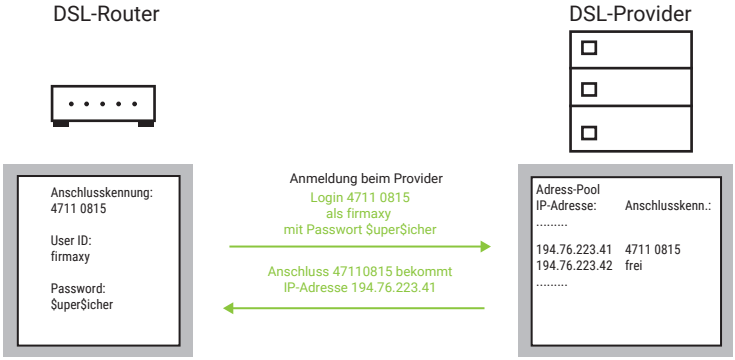
2. Verwendung von Dynamischem DNS
Einer der ersten Anbieter für dynamisches DNS war die Organisation DynDNS. In der Vergangenheit konnte man bei DynDNS nach einmaliger Anmeldung kostenlos einen weltweit einmaligen Hostnamen registrieren lassen.

Heute ist dieser Dienst leider kostenpflichtig.

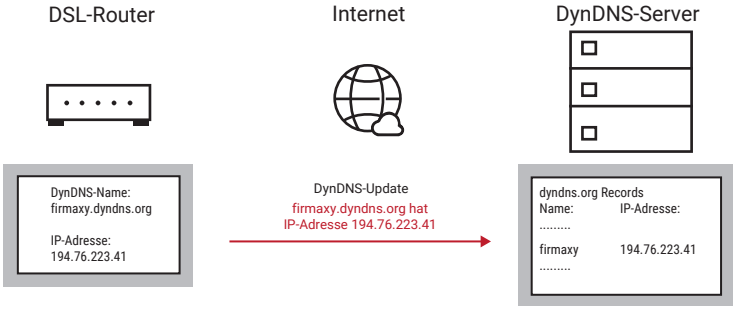
Eine detaillierte Beschreibung der Vorgehensweise ist auf den Webseiten von DynDNS unter http://www.dyndns.org verfügbar.

Die Abwicklung der Adressauflösung mittels DynDNS erfolgt in drei Schritten.

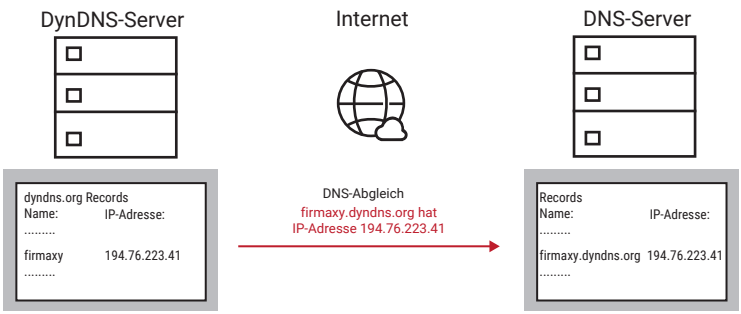
1. Der Internet-User stellt z.B. via DSL eine Verbindung zu seinem Internet-Provider her und bekommt nach erfolgreichem Login eine IP-Adresse zugeteilt.



2. Im Gegensatz zu DDNS muss der Anwender bzw. sein Endgerät dafür sorgen, dass DynDNS weiß, unter welcher IP-Adresse das Endgerät erreichbar ist. Dazu nutzt das Endgerät den DynDNS-Update-Client. Für PCs gibt es entsprechende Programme, die diese Aufgabe übernehmen. Bei anderen Endgeräten müssen spezielle Funktionen integriert sein. Bei Zugang zum Internet über einen Router übernimmt dieser meist auch das DynDNS-Update.



3. Erfolgt nun bei einem DNS-Server die Anfrage nach dem vom Internet-User benutzten DynDNS-Namen und der zugehörigen IP-Adresse, fragt der zuständige DNS-Server diese beim DynDNS-Server an und gleicht seine Daten ab.

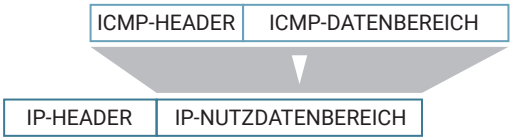


Damit ist das Endgerät unter dem gewählten Namen weltweit ansprechbar, kann also auch Server-Dienste anbieten.

ICMP – Erreichbarkeit prüfen mit Ping

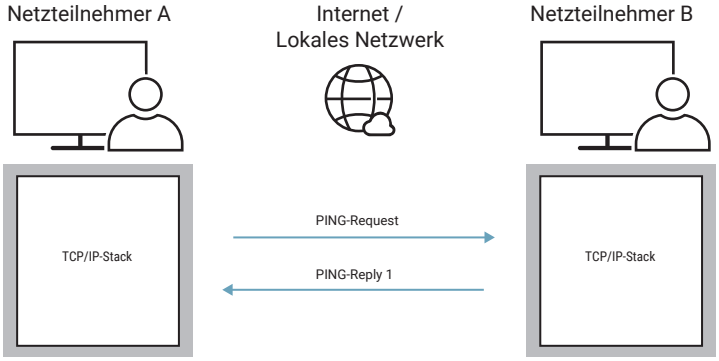
Die Ping-Funktion dient in TCP/IP-Netzen zu Diagnosezwecken. Mit Hilfe von Ping lässt sich überprüfen, ob ein bestimmter Teilnehmer im Netz existiert und tatsächlich ansprechbar ist.

Ping arbeitet mit dem ICMP-Protokoll (Internet Control Message Protocol), welches auf das IP-Protokoll aufsetzt.



Das Paket sieht dann so aus:

Setzt ein Netzteilnehmer durch Eingabe des Ping-Kommandos einen ICMP-Request ab, gibt die angesprochene Station einen ICMP-Reply an den Absender zurück.



Der Aufruf des Kommandos *PING <IP-Adresse>* in der DOS-Box fordert den durch die IP-Adresse angegeben Netzteilnehmer auf, eine Rückmeldung zu geben.

Zusätzlich können unter Windows noch diverse Parameter angegeben werden:

-t
Wiederholt das *Ping*-Kommando in Dauerschleife, bis der Anwender mit *<Strg> C* unterbricht.

-n count
Wiederholt das Ping-Kommando *count* mal.

-l size
size gibt an, mit wieviel Bytes das ICMP-Packet aufgefüllt wird. Bei Com-Servern in Default-Einstellung sind dies maximal 560 Bytes.

-w timeout
timeout spezifiziert, wie lange (in Millisekunden) auf die Rückmeldung gewartet wird.

Ein Beispiel:

```
PING 172.16.232.49 -n 50
```

sendet 50 *Ping*-Kommandos an die Station 172.16.232.49. Ist der Netzteilnehmer vorhanden, erscheint folgende Rückmeldung:

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

Bleibt die Rückmeldung aus, wird eine entsprechende Meldung zurückgegeben:

```
Request timed out.
```

Anstelle der IP-Adresse kann natürlich auch ein Host-Name eingegeben werden. Die Voraussetzung hierzu ist der Zugang zu einem DNS-Server.

Die von Ping verwendeten ICMP-Pakete sind im Internet-Standard RFC-792 definiert.

Anwendungsprotokolle

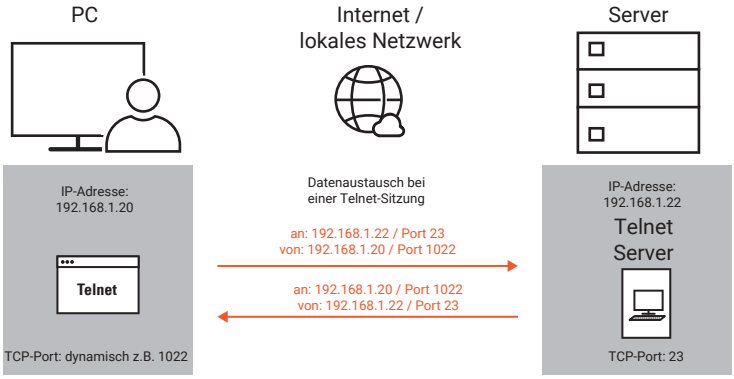
Anwendungsprotokolle verrichten eine für den Anwender sofort erkennbare Aufgabe oder können direkt durch den Anwender benutzt werden, schaffen also eine Schnittstelle zum Nutzer.

Im Anschluss an die oben genannten Hilfsprotokolle gehen wir in diesem Kapitel noch auf die folgenden Anwendungsprotokolle näher ein:

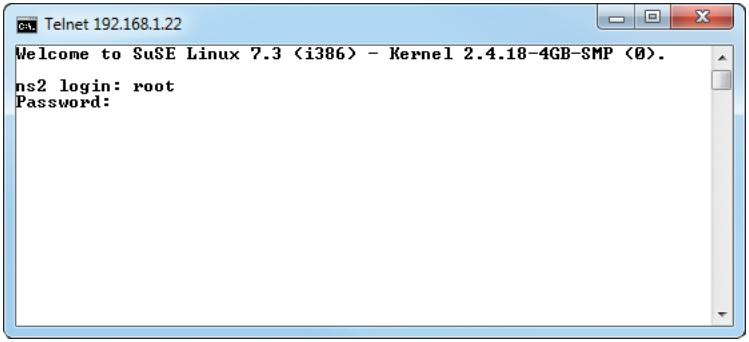
- Telnet
- FTP
- TFTP
- SNMP
- Syslog

Telnet - Terminal over Network

Einfach ausgedrückt ist Telnet ein Textfenster bzw. textorientiertes Programm, über das ein anderer Rechner (Telnet-Server) im Netzwerk vom Anwender fernbedient werden kann.



Eine Telnet-Sitzung kann man sich vorstellen wie eine DOS-Box, allerdings werden die eingetippten Befehle auf dem entfernten Rechner ausgeführt.



Dafür werden mehrere Elemente benötigt.

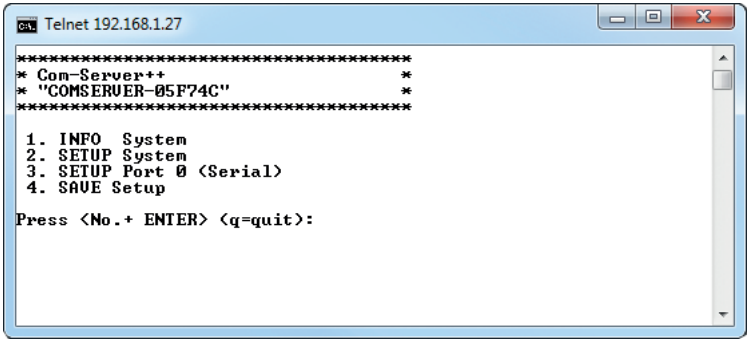
Der Telnet-Client

Alle modernen Betriebssysteme verfügen heute über ein Telnet-Client-Programm. Bei Windows7 bzw. Windows10 muss der Telnet Client allerdings erst aktiviert werden. Das erfolgt in der Systemsteuerung über *Programme und Funktionen >> Windows-Funktionen (Features) aktivieren oder deaktivieren >> Telnet-Client*. Alternativ können aber auch Telnet-Clients von Drittanbietern, wie z.B. Putty, genutzt werden.

Der Telnet-Client baut eine TCP-Verbindung zu einem Telnet-Server auf, nimmt Tastatureingaben vom Anwender entgegen, gibt sie an den Telnet-Server weiter und stellt umgekehrt die vom Server gesendeten Zeichen auf dem Bildschirm dar.

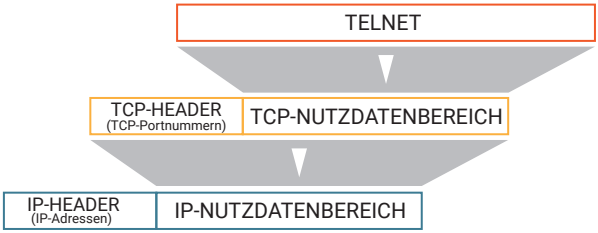
Der Telnet-Server

Der Telnet-Server ist auf dem entfernten Rechner aktiv und gibt einem oder ggf. mehreren Nutzern die Gelegenheit, sich dort „einzuloggen“. Damit ist der Telnet-Server (in Unix-Systemen auch oft als Telnet-Daemon bezeichnet) das Bindeglied zwischen Netzwerkzugang via Telnet-Client und dem zu bedienenden Prozess. In seinem Ursprung wurde Telnet eingesetzt, um einen Remote-Zugang zu Unix-Betriebssystemen zu schaffen. Es verfügen auch viele Embedded Systeme wie Com-Server oder Printer-Server, Switches, Hubs und Router über einen Telnet-Server, der als Konfigurationszugang dient.



Das Telnet Protokoll

Auch Telnet setzt auf TCP als Basisprotokoll auf.



Das Telnet-Datenpaket sieht dann so aus:

Hierbei wird, wenn vom Anwender nicht anders vorgegeben, der Port 23 genutzt. Es kann aber auch jeder beliebige andere Port angegeben werden. Wichtig ist, dass auf dem gewählten Port ein Telnet-Server aktiv ist. Das Telnet-Protokoll übernimmt im Wesentlichen drei Aufgaben:

1. Festlegung benutzter Zeichensätze und Steuercodes zur Cursor-Positionierung

   Als gemeinsame Basis für Client und Server wird hierzu der NVT-Standard „Network Virtual Terminal“ eingesetzt. NVT benutzt den 7Bit-ASCII-Zeichensatz und legt fest, welche Zeichen dargestellt werden und welche zur Steuerung und Positionierung genutzt werden.

2. Aushandeln und Einstellen von Verbindungsoptionen

   Über die Festlegungen im NTV hinaus kann Telnet von einer Vielzahl spezieller Funktionen Gebrauch machen. Das Telnet-Protokoll gibt Client und Server die Möglichkeit, Verbindungsoptionen auszuhandeln. Zum Beispiel: ob der Server alle vom Client empfangenen Zeichen als Echo zurückgeben soll.

   Hierzu werden Steuerzeichen benutzt, bei denen das achte Bit gesetzt ist, also Zeichen oberhalb 127 und damit außerhalb des NTV-Zeichensatzes.

3. Transport der Zeichen, die zwischen Client und Server ausgetauscht werden

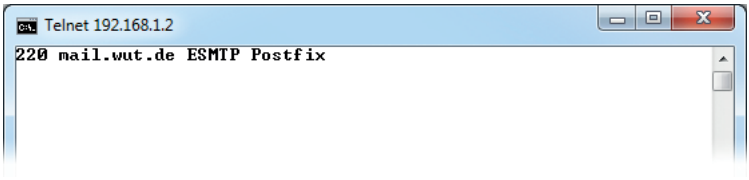
   Alle vom Anwender eingegebenen oder vom Server gesendeten Zeichen des NTV-Zeichensatzes werden 1:1 in den Nutzdatenbereich eines TCP-Paketes gepackt und über das Netzwerk transportiert.

Die Einfachheit des Telnet-Protokolls sowie die Transparenz bei der Zeichenübertragung haben Telnet auch zu einem beliebten Diagnosetool gemacht. So lassen sich Verbindungen zu HTTP- , SMTP- oder POP3-Servern herstellen.

Es lässt sich zum Beispiel durch Eingabe der folgenden Zeile in einer Dos-Box überprüfen, ob der SMTP-Server (Port 25) arbeitet:

telnet <IP-Adresse eines Mailservers> 25

Ist der SMTP-Server aktiv, wird eine Begrüßungsmeldung zurückgegeben.



Durch konsequentes Eintippen des SMTP-Protokolls könnte man nun theoretisch per Telnet-Client E-Mails verschicken.

FTP - File Transfer Protocol

In einfachen Worten ausgedrückt, erlaubt FTP einem Anwender im Netzwerk den Zugriff auf das Datei-System bzw. die Festplatte eines entfernten Rechners.

Eine der Hauptanwendungen für FTP ist heute das Aufspielen von HTML-Seiten auf WWW-Server, die zu diesem Zweck auch immer einen FTP-Zugang haben.

FTP kann aber auch genutzt werden, um über embedded FTP-Clients, wie zum Beispiel den W&T Com-Server, serielle Daten von Endgeräten in eine Datei auf dem Server zu speichern.

Ein weiteres Anwendungsfeld ist das Data-Logging (zyklisches Abspeichern von Datensätzen) via FTP. Auf diesem Weg kann z.B. ein W&T Web-Thermometer die Werte für Temperatur und Luftfeuchte in vorgegebenen Abständen mit Zeitstempel in eine Datei auf dem FTP-Server schreiben.

Der FTP-Client

FTP arbeitet nach dem Client/Server-Prinzip. Ein FTP-Client ist heute Bestandteil jedes Betriebssystems. Unter Windows z.B. wird durch Eingabe des FTP-Befehls in einer Dos-Box der FTP-Client gestartet.

Mit dem *OPEN*-Kommando, gefolgt von der IP-Adresse bzw. dem Hostnamen des FTP-Servers, wird die FTP-Verbindung geöffnet und der Nutzer muss seinen Login-Namen und ein Passwort eingeben. Nach erfolgreichem Login sind je nach Zugriffsrecht unter anderem folgende Dateioperationen möglich:

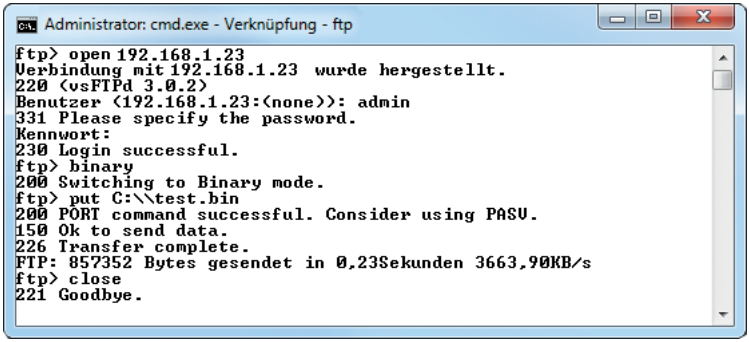
	FTP Befehl
Speichern von Dateien auf dem Server	PUT
Laden von Dateien vom Server	GET
Daten an eine bestehende Datei anhängen	APPEND
Löschen von Dateien auf dem Server	DELETE
Anzeigen des Verzeichnisinhaltes	DIR

Eine Auflistung aller unterstützten Kommandos erhält man mit der Eingabe eines „*?*“ hinter dem FTP-Prompt. Eine kurze Beschreibung der einzelnen Kommandos kann mit „*? Kommando*“ abgerufen werden.

Eine wichtige Eigenschaft von FTP ist die unterschiedliche Handhabung von Text- und Binärdateien. Um die gewünschte Betriebsart auszuwählen, stellt FTP zwei weitere Kommandos zur Verfügung:

	FTP Befehl
für die Übertragung von Textdateien	ASCII
für die Übertragung von Binärdateien (z.B. ausführbare Programmdateien)	BINARY

Nach der Eingabe von FTP findet die Bedienung in einer Art Dialog statt, wie hier beispielhaft für das Speichern der Datei *test.bin* auf dem Server *192.168.1.23* gezeigt:



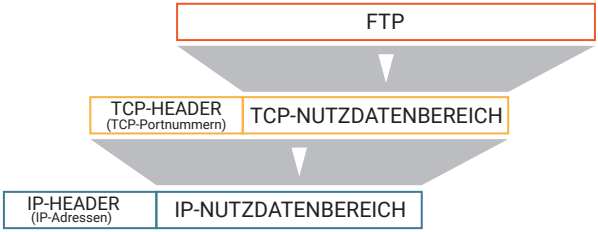
Je nach Betriebssystem können sowohl die Bedienung als auch die Kommandos des FTP-Client variieren.

In Unix-Betriebssystemen ist außerdem strikt auf Groß- und Kleinschreibung zu achten.

Eine komfortablere Handhabung von FTP lässt sich durch den Einsatz von zugekauften FTP-Client-Programmen mit grafischer Benutzeroberfläche erreichen.

Das FTP-Protokoll

Als Basis-Protokoll setzt FTP auf das verbindungsorientierte und gesicherte TCP auf.



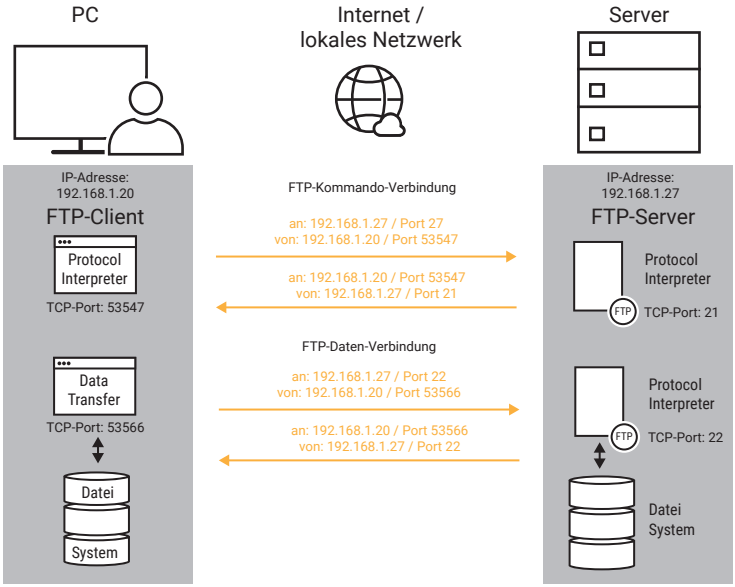
Das FTP-Datenpaket sieht dann so aus:

Im Gegensatz zu anderen Internetdiensten nutzt FTP aber zwei TCP-Verbindungen und damit zwei TCP-Ports:

- Port 21 als Kommando-Verbindung
- der zweite Port wird für die Übertragung der Dateien benutzt. Die verwendete Portnummer wird ausgehandelt.

Die Steuerung des Dateitransfers zwischen Client und Server wird über einen Kommandodialog gesteuert. Diesen Part wickeln die Protocol-Interpreter über die Kommandoverbindung ab. Die Kommandoverbindung bleibt für die gesamte Dauer der FTP-Sitzung bestehen.

Der eigentliche Dateitransfer erfolgt über die Datenverbindung, die vom Data-Transfer-Prozess für jede Dateioperation neu geöffnet wird. Der Data-Transfer-Prozess ist dabei das Bindeglied zwischen Netzwerk und Dateisystem und wird vom Protocol-Interpreter gesteuert.



Der FTP-Server

Ein FTP-Server steht in der Regel nur bei Server-Betriebssystemen zur Verfügung und muss ggf. erst gestartet werden.

FTP-Server bieten zwei Zugriffsmöglichkeiten:

1. Nur eingetragene Nutzer haben Zugriff und können, je nach in einer User-Liste festgehaltenem Zugriffsrecht, Dateioperationen ausführen.
2. Jeder Nutzer kann auf den Server zugreifen. Ein Login findet entweder gar nicht statt oder es wird der Username „anonymous“ angegeben. Man spricht dann von Anonymous-FTP.

Passives / Aktives FTP

Bei FTP unterscheidet man zwei Arten der Abwicklung von TCP-Verbindungen.

1. **Passives FTP**
   Sowohl für die Kommando-Verbindung als auch für die Datenverbindung baut der FTP-Client jeweils die TCP-Verbindung auf - er fungiert aus TCP-Sicht also als Client für beide Verbindungen.
2. **Aktives FTP**
   Der FTP-Client baut die Kommandoverbindung zum FTP-Server auf. Gleichzeitig startet der FTP-Client einen Server-Prozess für den Datenaustausch. Dazu übermittelt er dem FTP-Server die TCP-Portnummer, auf der er die Datenverbindung entgegennehmen möchte. Für die Datenverbindung fungiert also der FTP-Client als Server und der FTP-Server als Client.

Aktives FTP und Firewalls

Bei netzwerkübergreifendem FTP kann es zu Problemen führen, wenn das FTP-clientseitige Netzwerk durch eine Firewall geschützt ist.

Zur Erinnerung: In aller Regel sind Firewalls so konfiguriert, dass Verbindungen aus dem lokalen Netz ins öffentliche Netz (Internet) erlaubt sind. Verbindungen aus dem öffentlichen Netz ins lokale Netz müssen aber explizit freigegeben werden.

Oft blocken Firewalls die eingehende Datenverbindung beim aktiven FTP. In solchen Fällen sollte passives FTP verwendet werden.

Zwar gibt es Firewalls, die aktives FTP erkennen und automatisch den benötigten Port für die Dauer der Datenverbindung freigeben - das wird aber nicht von allen Firewalls unterstützt.

TFTP - Trivial File Transfer Protocol

Neben FTP ist TFTP ein weiterer Dienst, um über das Netzwerk auf die Dateien eines entfernten Rechners zugreifen zu können.

TFTP ist allerdings sowohl vom Funktionsumfang als auch von der Größe des Programmcodes deutlich „schlanker“ als FTP.

Ein TFTP-Client ist nicht unbedingt Bestandteil des Betriebssystems.

TFTP-Server kommen im Officebereich selten zum Einsatz.

Besonders geeignet ist TFTP für den Einsatz in Embedded Systemen, in denen nur begrenzter Speicherplatz für Betriebssystemkomponenten zur Verfügung steht. TFTP bietet hier bei minimalem Programmcode ein hohes Maß an Effizienz.

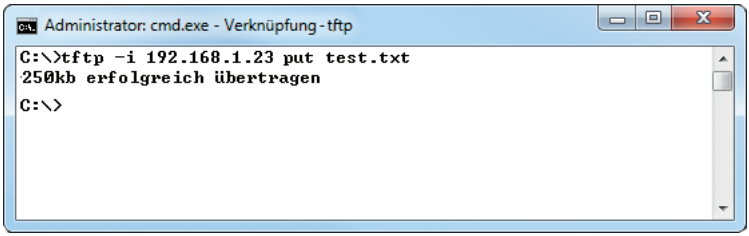
In Com-Servern, Printerservern und Miniterminals wird beispielsweise TFTP genutzt, um Konfigurations- und Firmware-Dateien zu übertragen.

TFTP stellt nur zwei Dateioperationen zur Verfügung:

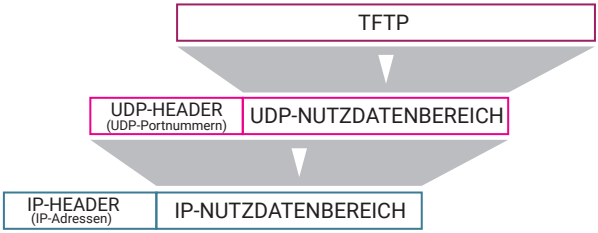
	TFTP Befehl
Speichern von Dateien auf dem Server	PUT
Laden von Dateien vom Server	GET

Wie auch FTP unterscheidet TFTP zwischen der Übertragung von Text- und Binär-Dateien. Sollen Binär-Dateien übertragen werden, wird dies durch den zusätzlichen Parameter „-i“ angegeben.

Hier als kurzes Beispiel: Die binäre Datei „test.txt“ wird von einem Windows Rechner auf den Server mit der IP-Adresse 192.168.1.23 gespeichert.



Auf eine Authentifizierung, also ein Login mit Passwortabfrage wie bei FTP, wird verzichtet. Im Gegensatz zu FTP verwendet TFTP als Basisprotokoll UDP, wobei der Port 69 genutzt wird.

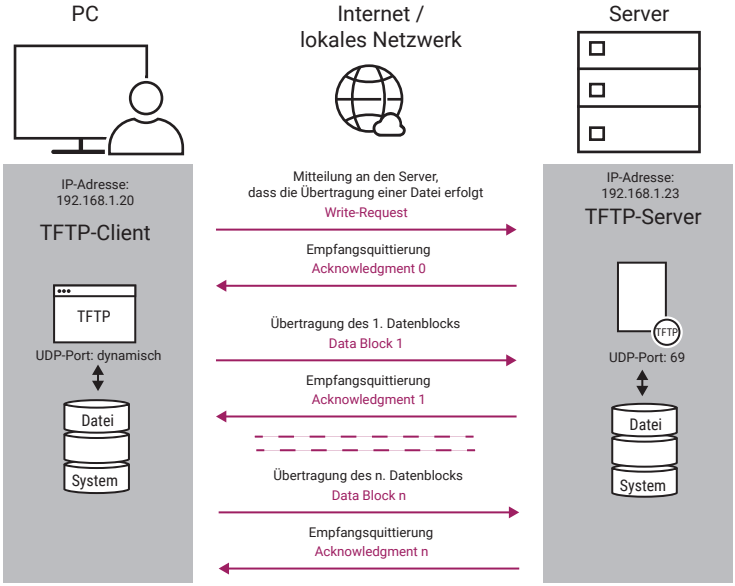


Das TFTP-Datenpaket sieht dann so aus:

Zur Erinnerung:
UDP arbeitet verbindungslos. Man spricht bei UDP-Paketen auch von Datagrammen, wobei jedes Paket als eigenständige Datensendung behandelt wird. Auf UDP-Ebene werden empfangene Pakete nicht quittiert. Der Sender erhält keine Rückmeldung, ob ein gesendetes Paket wirklich beim Empfänger angekommen ist. UDP-Pakete bekommen keine Sequenz-Nummer. Ein Empfänger, der mehrere UDP-Pakete erhält, hat keine Möglichkeit festzustellen, ob die Pakete in der richtigen Reihenfolge empfangen wurden.

Aus diesem Grund übernimmt TFTP die Sicherung der übertragenen Daten selbst.

Die Übertragung von Dateien geschieht in Blöcken von je 512Bytes, wobei die Blöcke mit einer laufenden Nummer versehen werden. Jeder empfangene Block wird von der Gegenseite quittiert. Erst nach Empfang der Quittung wird der nächste Block gesendet.



TFTP erkennt, ob die empfangenen Datenblöcke in Ordnung sind; eine Fehlerkorrektur gibt es aber nicht. Geht bei der Übertragung etwas schief, stimmt etwa die Paketlänge nicht oder ein komplettes Paket geht verloren, wird das Paket von der Gegenseite nicht quittiert. Bei ausbleibender Quittierung wird das Datenpaket einige Male erneut versandt. Bleibt die Quittierung dauerhaft aus, wird die Übertragung abgebrochen. In diesem Fall kann der Anwender oder eine intelligente Anwendungssoftware den Vorgang erneut starten.

SNMP – Simple Network Management Protocol

Netzwerke verbinden meist eine Vielzahl verschiedener Endgeräte unterschiedlicher Hersteller. Jeder Hersteller hat dabei seine ganz eigene Methodik, auf welche Weise die Geräte parametriert und überwacht werden.

So stellen einige Hersteller spezielle Managementprogramme für ihre Endgeräte zur Verfügung, andere bieten dem Benutzer bzw. Administrator eine Weboberfläche an, über die sich die Komponenten im Browser überwachen und konfigurieren lassen.

Kleinere Netzwerke lassen sich mit diesen Mitteln bequem einrichten, überwachen und warten.

In größeren Netzwerken, mit zum Teil mehreren 100 Netzwerkteilnehmern, wäre es allerdings sehr mühsam, jedes Gerät mit anderen Mitteln zu konfigurieren und zu überwachen. Hier bietet SNMP die Grundlage für ein einheitliches und überschaubares Netzwerkmanagement.

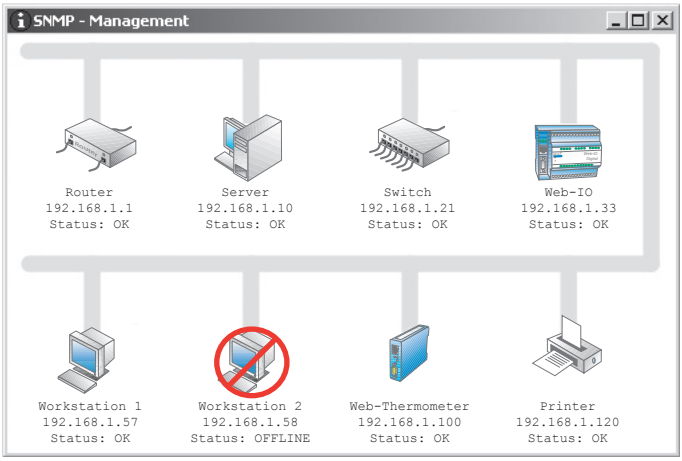
SNMP-Agent

Bedingung für den Einsatz von SNMP ist, dass alle beteiligten Endgeräte einen SNMP-Agenten besitzen. Der SNMP-Agent ist eine Software-Schnittstelle, die das Endgerät mit allen betriebswichtigen Parametern repräsentiert. SNMP-fähige Endgeräte werden auch als Netzwerkknoten bzw. Nodes bezeichnet. Nodes können Workstation-PCs, Server, Switches, Router, Web-IOs, also eigentlich alles sein, was über eine eigene IP-Adresse im Netzwerk ansprechbar ist.

SNMP-Manager

Neben den Nodes gibt es in SNMP-Systemen mindestens einen SNMP-Manager. Der SNMP-Manager ist eine Software-Anwendung, die auf einer Workstation oder einem Server arbeitet.

Während SNMP-Manager früher kommandozeilengesteuerte Anwendungen waren, in denen die Nodes in Listen verwaltet wurden, stellen moderne SNMP-Systeme dem Administrator mächtige Visualisierungsfunktionen zur Verfügung. Die gesamte Netzwerkinfrastruktur kann in Form von Plänen dargestellt und somit sehr übersichtlich verwaltet werden.



Zu den Aufgaben eines SNMP-Managers zählen: Konfiguration, Verwalten von Zugriffsrechten, Überwachen, Fehlermanagement und Netzwerksicherheit.

SNMP-MIB

Die Abkürzung MIB steht für Management Information Base. Zu jedem Netzwerkknoten gehört eine spezifische MIB, d.h. eine Liste aus abrufbaren Variablen, in denen die Eigenschaften und Zustände des Netzteilnehmers beschrieben sind.

Im Normalfall muss der Anwender sich nicht im Detail mit dem Aufbau der MIB beschäftigen. Moderne Managementsysteme verfügen über einen MIB-Compiler, der die MIB-Daten ins System integriert und dem Nutzer in einer gut handhabbaren Form zur Verfügung stellt.

Um das Verständnis für die Abläufe bei SNMP zu erhöhen, möchten wir hier dennoch einen groben Überblick über den Aufbau vermitteln.

Die MIB besteht aus zwei Teilen: der Standard-MIB, in der System-Variablen verwaltet werden, die für alle Knoten benötigt werden, und der Private-MIB, in der die gerätespezifischen Variablen untergebracht sind und auf die wir hier näher eingehen.

Die Datenstruktur der MIB hat einen baumartigen Aufbau, ähnlich der Verzeichnisstruktur auf einer Festplatte. Die einzelnen Variablen sind in Gruppen, Untergruppen usw. gegliedert, so wie einzelne Dateien auf einem Datenträger in Ordnern und Unterordnern gespeichert werden.

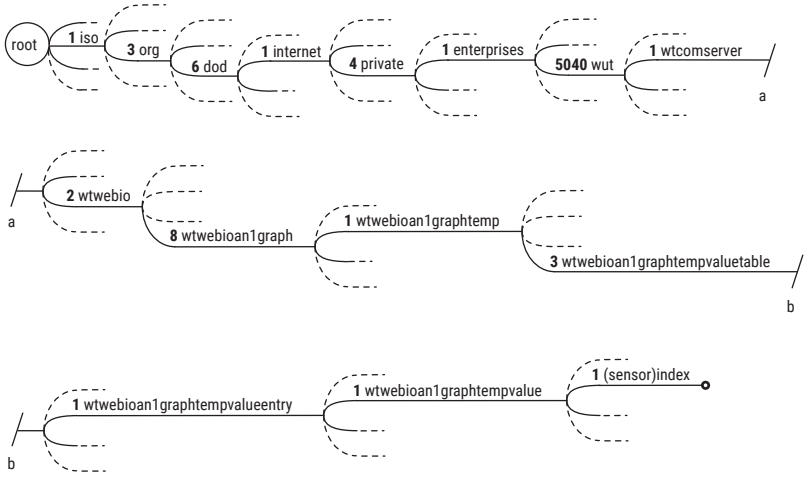
Die Abbildung zeigt in der Darstellungsweise eines Verzeichnisbaumes, an welcher Stelle zum Beispiel bei einem Web-Thermographen die gemessene Temperatur per SNMP abgerufen werden kann.

Bei den MIB-Variablen spricht man auch von Objekten. Zu jedem Objekt einer MIB gehört die MIB-OID. OID steht für Object Identifier. Die OID ist eine durch Punkte getrennte Kette von Zahlen, wobei jede Zahl für einen Abzweig im MIB-Baum angibt, wohin verzweigt wird.

Die OID für die Sensortemperatur des Wiesemann & Theis Web-Thermographen sieht z.B. so aus:

```
1.3.6.1.4.1.5040.1.2.8.1.3.1.1.1
```

Da solche Datenketten für den Anwender nicht überschaubar sind, kann man die OID auch als MIB-Diagramm darstellen:



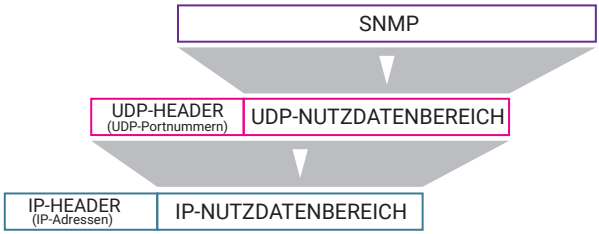
Die von den Herstellern der verschiedenen Netzwerkknoten mitgelieferten MIB-Dateien beschreiben die OID-Struktur im ASN.1-Format (Abstract Syntax Notification).

ASN.1-Dateien sind zwar lesbar, eine Entschlüsselung durch den Anwender ist aber kompliziert und nicht vorgesehen.

Wie bereits angesprochen, verfügen SNMP-Managementsysteme über einen ASN.1-MIB-Compiler. Dieser Compiler wertet das ASN.1-Format aus und vermittelt dem Manager, welche Variablen eines Netzwerkknotens an welcher Stelle zu finden sind.

SNMP-Kommunikation

Die Kommunikation zwischen SNMP-Managementsystem und SNMP-Netzknoten wird über das UDP-Protokoll abgewickelt.



Das SNMP-Datenpaket sieht dann so aus:

Hierbei empfängt der Netzwerkknoten die Datensendungen vom SNMP-Managementsystem auf Port 161.

Die normale Kommunikation geht immer vom Managementsystem aus. Dieses sendet ein GET-Kommando mit der OID des gewünschten Wertes an den Netzwerkknoten. Der Netzwerkknoten sendet daraufhin ein RESPONSE-Paket zurück, welches ebenfalls die OID und zusätzlich den zugehörigen Wert enthält. Dieses Frage-/Antwortspiel wird auch als Polling bezeichnet.

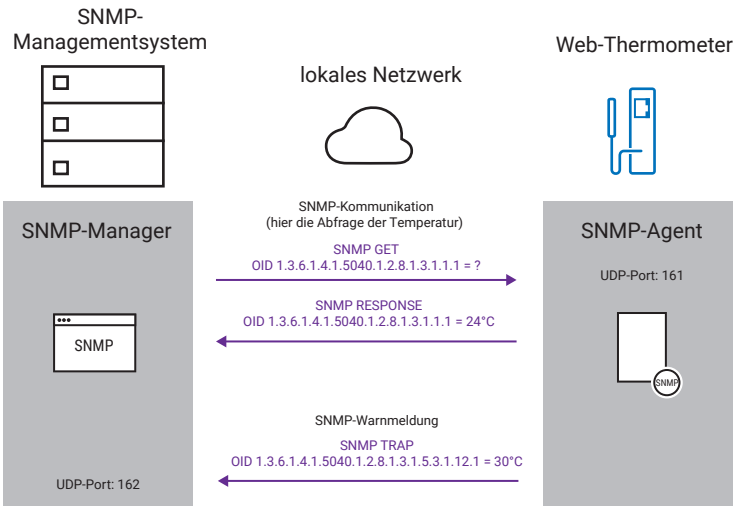
SNMP-Trap

Neben dem vom SNMP-Manager initiierten Polling gibt SNMP den Netzwerkknoten die Möglichkeit, unaufgefordert Meldungen an den SNMP-Manager zu senden.

Diese SNMP-Traps werden als Status- oder Warnmeldungen genutzt. So kann z.B. ein Switch auf diesem Weg melden, wenn ein Port seinen Link verliert, also das angeschlossene Endgerät nicht mehr erkannt wird.

SNMP-Traps werden an Port 162 gesendet.

Beim Web-Thermographen können Alarme definiert werden, die bei Temperaturüberschreitung (z.B. im Serverraum) einen SNMP-Trap senden.



SNMP-Traps haben eigene OIDs, die in einem gesonderten Teil der MIB untergebracht sind, auch wenn der gleiche Wert in einem anderen Teil der MIB ggf. noch einmal auftaucht.

Für Administratoren ausgedehnter Netze mit vielen Netzwerkteilnehmern bietet SNMP alle Voraussetzungen, die Wartung und Überwachung aller beteiligten Geräte einheitlich und übersichtlich abzuwickeln.

Community Strings

Der Community String ist eine Art Passwort, das bei jeder SNMP-Abfrage mitgesendet wird. SNMP sieht drei verschiedene Community Strings für verschiedene Zugriffsberechtigungen vor:

Read Only Community String für rein lesenden Zugriff
Read/Write Community String für lesenden und verändernden Zugriff
Trap Community String für das Versenden von SNMP-Traps

Die meisten SNMP-fähigen Geräte verwenden ab Werk in allen drei Fällen als Community String das Wort „public“. Die Community Strings sind aber frei konfigurierbar.

SNMP-Versionen

Inzwischen gibt es drei Versionen von SNMP:

SNMPv1

SNMPv1 ist die ursprüngliche Version von SNMP und umfasst bereits alle hier beschriebenen Funktionen. Ein Problem bei SNMPv1 ist die fehlende Sicherheit. Die ausgetauschten Daten gehen unverschlüsselt über das Netzwerk und können ggf. von Unberechtigten mitgelesen werden.

SNMPv2

Der wesentliche Unterschied zu SNMPv1 besteht darin, dass die Community Strings verschlüsselt übertragen werden. Darüber hinaus bietet SNMPv2 die Möglichkeit, in einer Tabelle zusammengefasste Daten komplett mit einem Abruf auszulesen.

SNMPv3

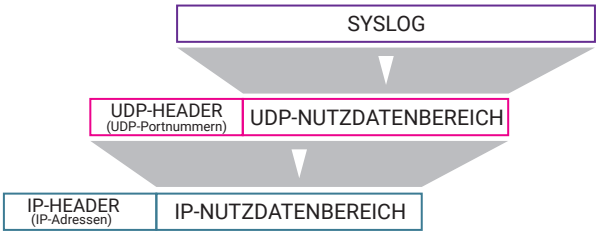
SNMPv3 ermöglicht die verschlüsselte Übertragung der Kommunikationsdaten. Außerdem wird mit Usernamen und Passwörtern gearbeitet.

Weitere Details zu verschlüsselter Datenübertragung im Kapitel Datensicherheit/Netzwerksicherheit.

Syslog - Der Systemlogger

Syslog ist ähnlich dem SNMP ein Protokoll, um Systemmeldungen an zentraler Stelle zu bündeln und zu überwachen. Im Gegensatz zu SNMP ist Syslog aber eine Einbahnstraße. Das heißt mit Syslog können Netzwerkgeräte wie PCs, Router, Switches, Hubs, aber auch Embedded Geräte wie Web-IO und Web-Thermographen, Systemmeldungen an einen zentralen Server senden; Datensendungen vom Server zu den Endgeräten sind jedoch nicht vorgesehen.

Auf Netzwerkebene werden Syslog-Meldungen über das UDP-Protokoll auf Port 514 übertragen.



Das SYSLOG-Datenpaket sieht dann so aus:

Syslog-Meldungen können normale Statusinformationen, Warnmeldungen und Fehlermeldungen sein.

Je nach Dringlichkeit werden den Syslog-Meldungen vom Absender Prioritäten zugeordnet. Auf diese Weise kann beeinflusst werden, welche Meldungen bevorzugt bearbeitet werden. Ferner enthält jede Syslog-Meldung einen Zeitstempel mit Uhrzeit und Datum.

Der Prozess, der auf dem Server die Syslog-Meldungen entgegennimmt und weiterverarbeitet, wird als Syslog-Daemon bezeichnet.

Syslog stammt im Ursprung aus der Unix- bzw. Linux-Welt, wird heute aber auch im Windows-Umfeld eingesetzt.

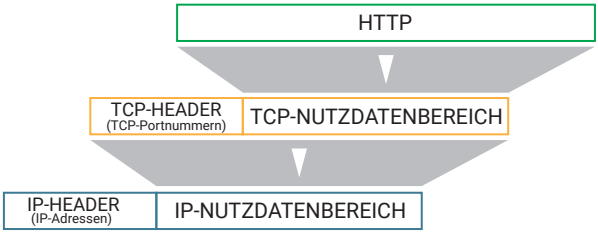
Web-Protokolle

Die beiden meistgenutzten Internet-Anwendungen sind das Abrufen von Webseiten im Browser und das Versenden von E-Mails. Die dazu notwendigen Protokolle sind:

- HTTP
- SMTP
- POP3
- IMAP

HTTP/HTTPS – Hypertext Transfer Protocol

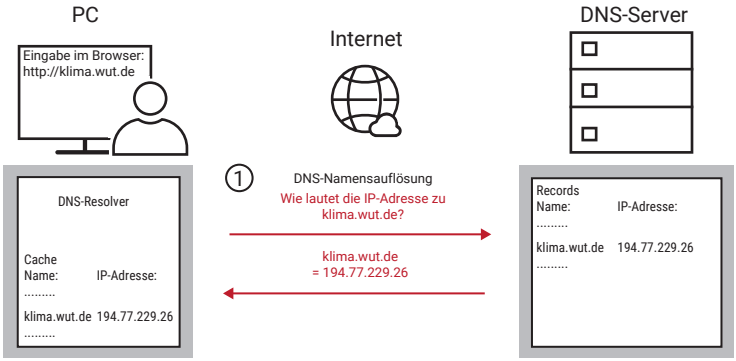
HTTP bzw. HTTPS ist das Protokoll, das der Browser nutzt, um Webseiten und andere Inhalte von Webservern abzurufen. HTTP setzt auf TCP als Basisprotokoll auf, wobei in aller Regel der TCP-Port 80 genutzt wird. Das S in HTTPS steht für Secure. HTTPS arbeitet grundsätzlich genauso wie HTTP, allerdings erfolgt die Übertragung verschlüsselt (dazu mehr im Kapitel Datensicherheit/Netzwerksicherheit). Der Standard-TCP-Port bei HTTPS ist 443. Abweichende Ports sind möglich, müssen aber explizit in der URL angegeben werden).



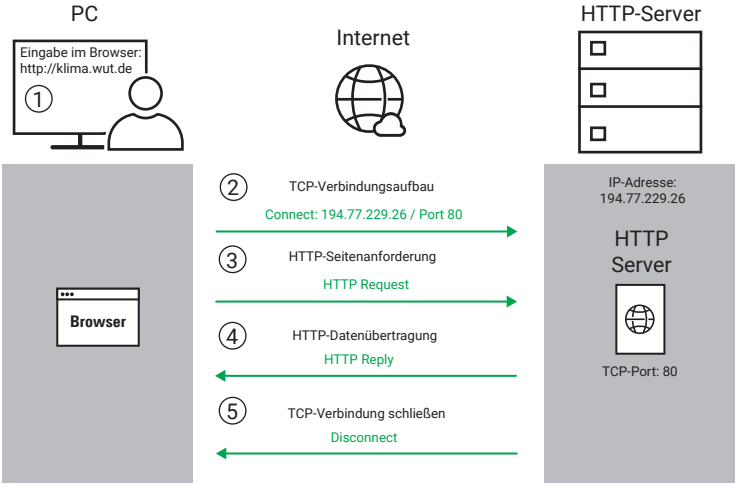
Das HTTP-Datenpaket hat somit folgenden Aufbau:

Die Anforderung und Übertragung einer Webseite erfolgt in fünf Schritten:

1. Auflösen des angegebenen Host- und Domainnamens in eine IP-Adresse
   Der TCP/IP-Stack startet eine DNS-Anfrage, um die IP-Adresse des gewünschten Servers zu ermitteln.



2. Aufbau der TCP-Verbindung
   Zur Erinnerung: Bei einer TCP-Verbindung gilt das Client/Server-Prinzip. Bei HTTP übernimmt der Browser die Rolle des Clients und stellt die TCP-Verbindung zum angegebenen HTTP-Server her.
3. Senden der HTTP-Anforderung
   Nach erfolgreichem Aufbau der TCP-Verbindung fordert der Browser die gewünschte Webseite beim HTTP-Server an. An dieser Stelle beginnt das eigentliche HTTP-Protokoll: Der Browser sendet das GET-Kommando mit den erforderlichen Parametern zum WWW-Server.
4. Senden der angeforderten Webseite
   Der HTTP-Server sendet erst eine HTTP-Bestätigung und dann die Webseite selbst.
5. Beenden der TCP-Verbindung durch den HTTP-Server



Eine Besonderheit bei HTTP ist, dass die TCP-Verbindung nicht wie sonst üblich durch den Client, sondern durch den Server abgebaut wird. Dafür gibt es zwei Gründe:

- Der HTTP-Server signalisiert dem Browser auf einfache Art und Weise, dass die Übertragung abgeschlossen ist.
- HTTP-Server müssen eine Vielzahl von TCP-Verbindungen gleichzeitig bedienen. Dabei verlangt jede offene Verbindung dem Server ein gewisses Maß an Leistung ab. Um die Verbindungszeiten so kurz wie möglich zu halten, baut der Server die Verbindung einfach ab, sobald alle angeforderten Daten übertragen wurden.

Die wichtigsten HTTP-Kommandos und -Parameter

Wie bereits angesprochen, basiert auch HTTP auf dem Client/Server-Prinzip: Der Browser als Client kann durch das Senden bestimmter Kommandos die Kommunikation steuern. Hier die beiden wichtigsten Kommandos:

Das GET-Kommando

Das mit Abstand am häufigsten verwendete Kommando ist die GET-Anfrage, die jeden Aufruf einer Webseite einleitet. GET fordert den HTTP-Server auf, ein Dokument oder Element zu senden und ist damit das wichtigste Kommando.

Für den Einsatz von GET sind einige Parameter nötig; man spricht auch von einer Kommandozeile (Request Line).

```
GET /pfadname/filename http-Version
```

Weitere Parameter können jeweils als neue Zeile mitgesendet werden. Diese angehängten Parameter werden auch als „Header“ bezeichnet.

Host	Hostname (nur bei HTTP1.1 nötig).
Accept	gibt an, welche Dateiformate der Browser verarbeiten kann Mit Accept: image/gif gibt der Browser z.B. bekannt, dass er Bilder im GIF-Format anzeigen kann.
Connection	über diesen Parameter (Connection: Keep-Alive) kann vom Browser vorgegeben werden, ob die TCP-Verbindung zum Nachladen anderer Elemente offengehalten wird.

Eine Vielzahl weiterer Parameter sind im RFC2616 beschrieben, der unter *https://www.w3.org/Protocols/rfc2616/rfc2616.html eingesehen werden kann.*

Ein typisches GET-Kommando könnte etwa so aussehen:

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

Als Antwort sendet der HTTP-Server eine Statuszeile, auf die ein Header (diesmal mit Parametern des Servers) folgt. Getrennt durch eine Leerzeile <CR LF CR LF> wird das angeforderte Element übermittelt.

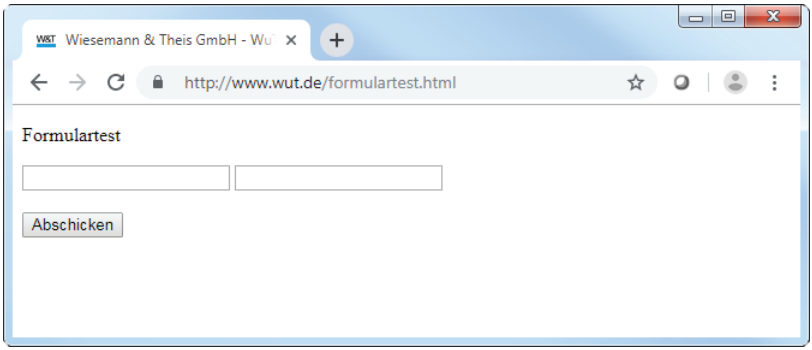
```
HTTP/1.1 200 OK                                | Statuszeile
Date: Thu, 15 Mar 2001 11:33:41GMT             |
Server: Apache/1.3.4 (Unix) PHP/3.0.6          |
Last-Modified: Thu 15 Mar 2001 11:32:32 GMT    |
...                                            |
...                                    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

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

Die Statuszeile umfasst die vom Server unterstütze HTTP-Version, eine Fehlercode-Nummer und einen Kommentar. Im Header zeigt der Server unterstützte Verbindungseigenschaften und Daten an.

Das POST-Kommando

Das Gegenstück zu GET ist das POST-Kommando. POST erlaubt dem Browser, Informationen an den HTTP-Server zu übergeben.

Der klassische Einsatz für das POST-Kommando ist die Übergabe von Formulareinträgen aus einer HTML-Seite. Im Kern ist der Aufbau der POST-Anforderung identisch mit der von GET. Nach den Parametern steht eine Leerzeile <CR LF CR LF>, der die zu übergebenden Informationen folgen. Enthält eine POST-Anforderung mehrere Einzelinformationen, werden diese durch ein „&“ voneinander getrennt. Als filename muss in der ersten Zeile der POST-Anforderung ein auf dem Server verfügbarer Prozess angegeben werden, der die Informationen entgegennehmen und verarbeiten kann.



Für dieses Formulartest-Formular könnte die POST-Anforderung folgendermaßen aussehen; der bislang nicht besprochene Parameter „Referer“ stellt hier einen Bezug zu der ursprünglich geladenen Formular-Seite her:

```
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

EINGABEFELD1=test1&EINGABEFELD2=test2&submit=Abschicken
```

Tipp: *Die meisten Internet-Provider bieten sogenannte „CGI-Scripts“ (Programme auf dem HTTP-Server) an, die Formularangaben entgegennehmen und als E-Mail an eine beliebige Adresse weiterleiten. So kann man seinen Kunden z.B. die Gelegenheit geben, direkt von einer Webseite aus eine Bestellung oder Anfrage zu verschicken.*

In der HTTP-Spezifikation sind weitere Kommandos definiert, die aber in der Praxis

so gut wie keine Bedeutung haben und auf die wir deshalb der Vollständigkeit wegen nur kurz eingehen:

- **HEAD**
  fordert wie GET eine Webseite an - der HTTP-Server liefert aber nur den HTTP-Kopf zurück. Suchmaschinen können mit HEAD prüfen, ob eine Webseite noch existiert.
- **PUT**
  dient zum Hochladen und (wenn bereits vorhanden) Ersetzen von Inhalten auf einen HTTP-Server.
- **PATCH**
  ändert bestehende Inhalte ohne sie vollständig zu ersetzen.
- **DELETE**
  löscht Inhalte auf einem HTTP-Server.
- **TRACE**
  liefert quasi ein Echo des gesendeten HTTP-Requests zurück. So kann geprüft werden, ob ein HTTP-Request auf dem Weg zum Server verändert wurde.
- **OPTIONS**
  liefert zurück, welche Methoden der angesprochene HTTP-Server unterstützt.

HTTP-Versionen

HTTP wurde seit der Einführung des WWW mehrfach weiterentwickelt und kommt heute in vier Versionen vor:

HTTP 0.9

1989 wurde HTTP 0.9 erstmalig vorgestellt und seitdem genutzt, aber nie spezifiziert.

HTTP 1.0

Erst 1996 wurde HTTP in der Version 1.0 durch die RFC 1945 spezifiziert, die weitestgehend mit HTTP 0.9 identisch ist.

HTTP 1.1

HTTP 1.1 wurde 1997 (RFC 2068) eingeführt und ist seit 1999 (RCF 2616) in überarbeiteter Form im Einsatz.

Die wohl grundlegendste Änderung in HTTP 1.1 liegt darin, dass die für die Übertragung des HTML-Dokuments aufgebaute TCP Verbindung auch für das Nachladen weiterer Elemente weiter genutzt wird. HTTP 1.0 bzw. 0.9 haben für jedes Element eine separate TCP-Verbindung aufgebaut. Eine persistente Verbindung

wie in 1.1 erhöht den Datendurchsatz, da die Zeiten für Verbindungsaufbau und -abbau entfallen.

Um auf einem HTTP-Server die Internet-Auftritte mehrerer Anbieter verwalten zu können, wurde mit „Host“ ein zusätzlicher Parameter zum GET-Kommando eingeführt, der dem Server zusammen mit einer GET-Anfrage auch den Hostnamen übermittelt (z.B Host: http://www.wut.de). Ein HTTP-Server hat nur eine IP-Adresse, auch wenn er mehrere Hostnamen repräsentiert. Dank dieses zusätzlichen Parameters kann der HTTP-Server über die GET-Anfrage erkennen, welchem Host die TCP-Verbindung gilt.

HTTP 2.0

Die offizielle Bezeichnung lautet HTTP/2. 2015 wurde HTTP/2 als Nachfolger von HTTP 1.1 mit folgenden Erweiterungen eingeführt:

- mehrere Anfragen können zusammengefasst werden
- schnellere Übertragung durch Datenkompression
- neben Text werden auch Binärdaten unterstützt
- der Server kann von sich aus Daten (ohne Anforderung) senden
- die Übertragung bestimmter Inhalte kann priorisiert werden

Damit ist HTTP/2 deutlich schneller und flexibler bei der Datenübertragung als seine Vorgänger.

Alle aktuellen Browser unterstützen standardmäßig HTTP/2, können aber auch problemlos mit Servern zusammenarbeiten, die HTTP 0.9, HTTP 1.0 oder HTTP 1.1 verwenden.

Browser-Cache und Proxy-Server

Wie bereits gesagt, ist HTTP eines der meistgenutzten Protokolle. Deshalb macht HTTP meist auch einen hohen Anteil der über das Internet übertragenen Daten aus.

Gerade bei Webseiten und deren Inhalten wie z.B. Bildern ist es so, dass durch wiederholten Aufruf der gleichen Webseite gleiche Inhalte immer wieder aufs Neue abgerufen werden.

Das bedeutet, dass gleiche Inhalte unnötig mehrfach aus dem Internet geladen werden.

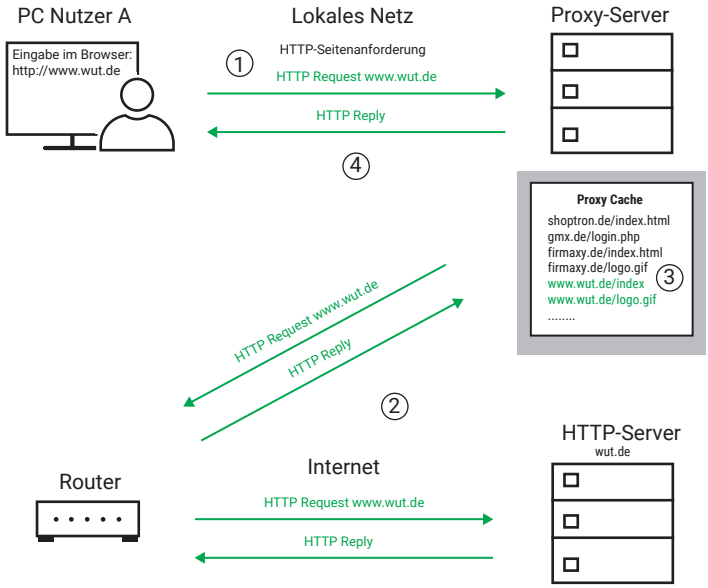
Um doppeltes Laden von Daten einzuschränken, bieten die gängigen Browser einen

sogenannten Cache - einen Zwischenspeicher, in dem geladene Inhalte vorübergehend abgelegt werden.

In den größeren Netzwerken von Firmen, Universitäten und anderen Einrichtungen sind zudem oft Proxy-Server im Einsatz. HTTP-Requests werden an den Proxy-Server umgelenkt, der ähnlich wie beim Browser-Cache einen Speicher hat, in dem die abgerufenen Inhalte vorübergehend abgelegt werden.

Während der Browser-Cache nur die von einem Nutzer abgerufenen Inhalte speichert, hält der Proxy-Server die von allen Nutzern abgerufenen Inhalte vor.

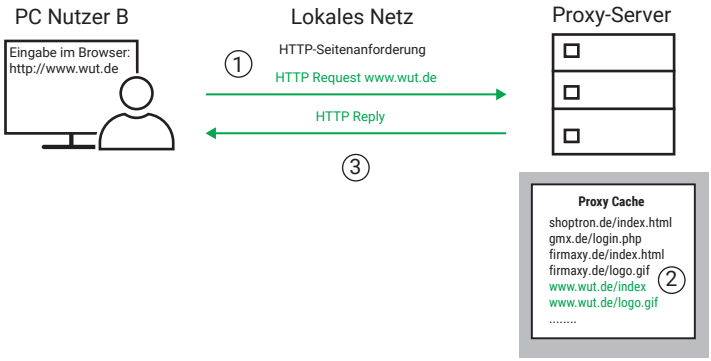
Beispiel:
Nutzer A ruft die Webseite von Wiesemann & Theis auf



1. Der HTTP-Request http://www.wut.de wird an den Proxy-Server gesendet.
2. Der Proxy-Server stellt fest, dass er die Inhalte von www.wut.de noch nicht kennt, und leitet den Request weiter an den HTTP-Server, auf dem www.wut.de gehostet wird. Der HTTP-Server sendet die angeforderten Inhalte an den Proxy.
3. Der Proxy-Server speichert die Inhalte von www.wut.de in seinem Cache.

4. Der Proxy-Server beantwortet den HTTP-Request des Browsers mit den Inhalten des HTTP-Servers.

Später ruft Nutzer B die Webseite von Wiesemann & Theis ebenfalls auf



1. Der HTTP-Request http://www.wut.de wird an den Proxy-Server gesendet.
2. Der Proxy-Server stellt fest, dass er die Inhalte von www.wut.de bereits in seinem Cache hat.
3. Der Proxy-Server beantwortet den HTTP-Request des Browsers mit den Inhalten aus seinem Cache.

In diesem Fall werden keine Daten über das Internet übertragen.

So kann durch den Browser-Cache und den Einsatz von Proxy-Servern das Datenvolumen über den Internetzugang reduziert werden.

Aktuelle Daten abrufen

Es gibt aber auch Webseiten-Aufrufe, bei denen es nicht gewünscht ist, ggf. veraltete Daten aus einem Cache zu bekommen. Was den Browser-Cache angeht, kann der Nutzer durch die Tastenkombination <Steuerung + F5> den Browser anweisen, die gewünschte Webseite aktuell vom Server abzurufen.

Bei Webseiten, die es zwingend erforderlich machen, bei jedem Aufruf aktuelle Daten zu liefern, kann das über entsprechende Header-Einträge, also Vorgaben im Kopf einer Webseite vorgegeben werden.

E-Mail

Die Möglichkeit, elektronische Post in wenigen Sekunden von einem Ende der Welt zum anderen verschicken zu können, ist sicherlich einer der Hauptgründe für die rasante Ausbreitung des Internets.

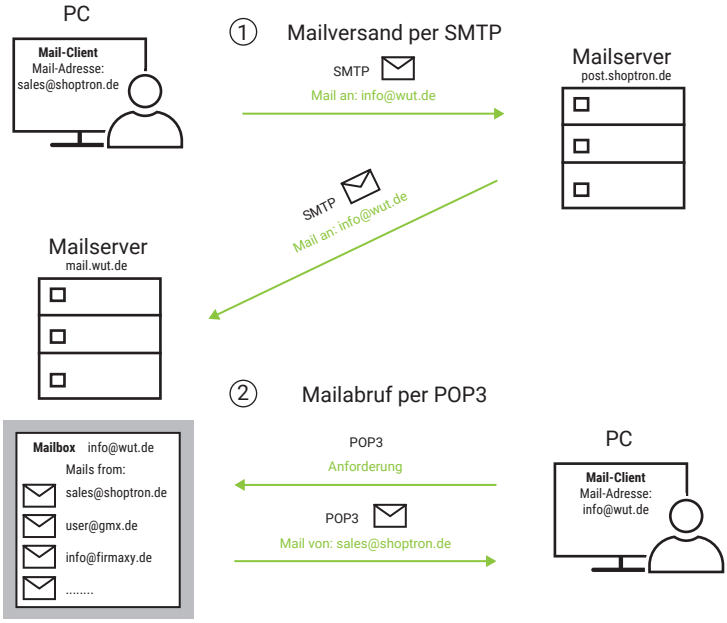
Im Gegensatz zu den meisten anderen Anwendungen im Internet ist das Versenden von E-Mails ein Dienst, bei dem keine direkte Verbindung zwischen Sender und Empfänger besteht. Das klingt zunächst verwirrend, ist aber sinnvoll, da sonst der Austausch von E-Mail nur möglich wäre, wenn Versender und Empfänger gleichzeitig im Netz aktiv sind.

Um eine zeitliche Unabhängigkeit zu gewährleisten, benötigt der E-Mailempfänger eine Mailbox (Postfach) auf einem Mailserver, in der eingehende Nachrichten zunächst abgelegt werden.

Eine E-Mail-Adresse setzt sich immer aus dem Postfachnamen und der Zieldomain zusammen; als Trennzeichen steht das „@“ (engl. „at“, gesprochen „ätt“) zwischen diesen beiden Bestandteilen. Ein Beispiel: info@wut.de bezeichnet das Info-Postfach auf dem Mailserver von Wiesemann & Theis.

Der Weg einer E-Mail vom Versender zum Empfänger besteht aus zwei Teilabschnitten, auf denen der Transport über unterschiedliche Protokolle geregelt wird:

1. Vom Rechner des Absenders bis zum Postfach des Empfängers wird das SMTP-Protokoll benutzt.
2. Vom Postfach des Empfängers bis zum Rechner des Empfängers wird das POP3-Protokoll benutzt.



Aufbau einer E-Mail

Eine E-Mail setzt sich aus dem Nachrichtenkopf und der eigentlichen Nachricht zusammen. Diesen Kopf kann man mit einem Briefumschlag vergleichen, der Felder für Absender, Empfänger, Datum, Betreff und einige weitere Informationen enthält.

Hier die wichtigsten Felder im Überblick:

Die folgenden fünf Felder bilden einen Minimalkopf und müssen auf jeden Fall enthalten sein.

Feld	Funktion
FROM	E-Mail-Adresse des Versenders
TO	E-Mail-Adresse des Empfängers
DATE	Datum und Uhrzeit Hinweis: die Uhrzeit kann willkürlich eingetragen werden und ist in aller Regel die Ortszeit des Absenders

Feld	Funktion
SUBJECT	Text der Betreffzeile
RECEIVED	Das Feld RECEIVED stellt eine Besonderheit dar, denn es wird nicht bei Erstellung der E-Mail angelegt. Jeder auf dem Weg der E-Mail liegende Mail-Router fügt ein RECEIVED-Feld ein und hinterlässt auf diese Weise einen „Durchgangsstempel“ mit Datum und Uhrzeit.

Die Verwendung der im Folgenden genannten Felder ist optional.

Feld	Funktion
SENDER	E-Mail-Adresse des Absenders (in aller Regel identisch mit Eintrag unter FROM)
REPLY-TO	E-Mail-Adresse, an die der Empfänger antworten soll. Wichtig, wenn E-Mails von einem Embedded System wie dem W&T Web-IO automatisiert verschickt werden. Als Antwortadresse könnte in diesem Fall z.B die E-Mail-Adresse des Administrators eingetragen sein.
CC	E-Mail-Adresse eines weiteren Empfängers, der einen „Durchschlag“ (CC = „Carbon Copy“) der Nachricht erhält.
BCC	E-Mail-Adresse eines weiteren Empfängers, die für alle anderen Empfänger aber unsichtbar bleibt (BCC = „Blind Carbon Copy“).
MESSAGE-ID	Eindeutige Identifikation einer E-Mail, die von der Mailsoftware willkürlich vergeben wird.
X-“MEINFELD“	Durch Voranstellen von „X-“ können eigene Felder erzeugt werden.

Bei einigen Feldern ist eine RESENT-Variante möglich, die dann zum Tragen kommt, wenn es sich um eine vom ursprünglichen Empfänger weitergeleitete E-Mail handelt.

Der formale Aufbau von Nachrichtenkopf und Feldern muss den folgenden Konventionen genügen:

- Nach dem Feldnamen steht ein Doppelpunkt; es folgt der jeweilige Parameter.

- Jedes Feld steht in einer eigenen Zeile, die mit <CR LF> (Carriage Return Line Feed; hex 0D 0A) endet.
- Nachrichtenkopf und -körper werden durch eine zusätzliche Leerzeile <CR LF> getrennt.
- Der Nachrichtenkörper selbst enthält nur den zu übermittelnden Text bzw. weitere eingefügte Dateien. Das Ende der Nachricht wird durch <CR LF . CR LF> (hex 0D 0A 2E 0D 0A) gekennzeichnet.
- Sowohl Kopf als auch Nachrichtenkörper bestehen ausschließlich aus 7-Bit-ASCII-Zeichen. Deshalb können auch alle Steuerinformationen als Klartext übertragen werden.

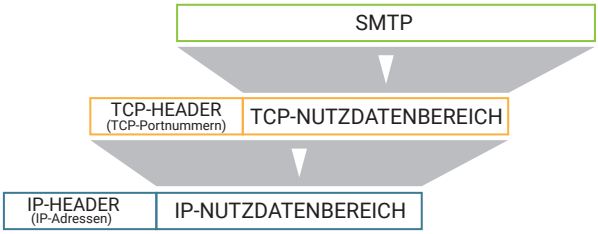
MIME – Multipurpose Internet Mail Extensions

Um auch binäre Daten (8-Bit-Format) via E-Mail verschicken zu können, werden diese vor dem Einbinden in den Nachrichtenkörper nach dem „MIME-Standard" in das 7-Bit-Format codiert und beim Empfänger wieder decodiert. Da die Verarbeitung binärer Daten von heutigen Mailprogrammen automatisch übernommen wird, verzichten wir an dieser Stelle auf eine detaillierte Erklärung der „MIME-Codierung".

SMTP – Simple Mail Transfer Protocol

SMTP regelt den Versand von E-Mails vom Mailclient zum Mailserver (SMTP-Server). Der Mailclient kann dabei entweder der ursprüngliche Versender oder ein auf dem Weg liegender Mail-Router sein. Mail-Router kommen zum Einsatz, wenn die E-Mail auf ihrem Weg über mehrere Domains weitergereicht wird. Häufig findet man für Mail-Router auch die Bezeichnung MTA (Mail-Transfer-Agent).

Für jedes Teilstück, das eine E-Mail zurücklegt, wird eine eigene TCP-Verbindung aufgebaut. SMTP setzt auf diese TCP-Verbindung auf, wobei der TCP-Port 25 genutzt wird.



Der Aufbau des SMTP-Datenpaketes sieht somit aus wie folgt:

SMTP stellt einige Kommandos (z.B Angabe des Absenders, Angabe des Empfängers, ...) zur Verfügung. Jedes SMTP-Kommando wird einzeln vom SMTP-Server quittiert. Die eigentliche E-Mail wird komplett mit Kopf und Körper gesendet und dann erst vom SMTP-Server quittiert. Wenn keine weiteren E-Mails zum Versand anstehen, wird auch die TCP-Verbindung wieder abgebaut.

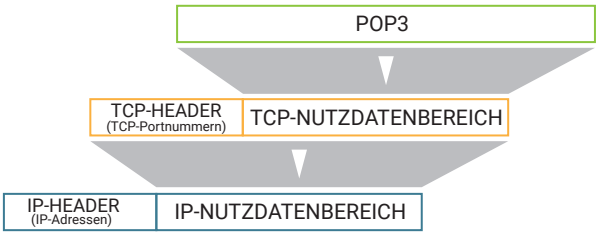
Hat die E-Mail den Ziel-Mailserver erreicht, wird sie im Postfach des Empfängers abgelegt und bleibt dort so lange liegen, bis sie vom Empfänger abgeholt wird.

POP3 – Post Office Protocol Version 3

Um eingegangene E-Mails aus dem Postfach auf dem Mailserver abzuholen, wird in den meisten Fällen das POP3-Protokoll benutzt. Der Empfänger wird über eingehende E-Mails nicht informiert. Er muss sein Postfach selbständig auf eingegangene E-Mails überprüfen und kann diese zu einem beliebigen Zeitpunkt abholen.

Die meisten der heute genutzten Mailprogramme überprüfen bei Start zunächst automatisch das Postfach des Nutzers auf eingegangene Mail. Viele E-Mail-Programme bieten darüber hinaus die Möglichkeit, ein Intervall vorzugeben, in dem das Postfach zyklisch geprüft wird. Typische Nutzer, die die meiste Zeit des Tages „offline“ sind, erhalten ihre E-Mails ohnehin nur dann, wenn sie sich beim Provider eingewählt haben. Doch bei Computern mit permanentem Internetzugang ist die zyklische Abfrage durchaus sinnvoll: Der Nutzer ist hier ständig online und erhält seine E-Mails mit nur geringer Verzögerung – quasi in Echtzeit.

Auch das POP3-Protokoll setzt auf eine TCP-Verbindung auf und ist nichts anderes als ein Klartextdialog, also ein Austausch lesbarer Kommandos.



Der Aufbau des POP3-Datenpaketes sieht somit wie folgt aus:

POP3 nutzt die TCP-Portnummer 110. Wie bei SMTP beginnt der Dialog auch hier mit einem Login. Bei POP3 muss sich der Empfänger allerdings in zwei Schritten anmelden: mit Nutzernamen und mit Passwort. Nach erfolgreichem Login stellt POP3 einige Kommandos zur Verfügung, mit denen eingegangene Nachrichten aufgelistet, abgeholt oder gelöscht werden können.

Heute wird der Nutzer mit SMTP und POP3 nur noch in geringem Maße konfrontiert: Er muss lediglich beim Einrichten der Mailsoftware den Namen des POP3- und SMTP-Servers angeben – das Abwickeln der Protokolle selbst wird unsichtbar im Hintergrund vom Mailprogramm übernommen.

Der Vollständigkeit halber sei noch erwähnt, dass es neben dem POP3-Protokoll noch die Protokolle POP2 und POP1 (beides Vorläufer von POP3) gibt, die ebenfalls zum Abholen von E-Mails entwickelt wurden. Diese Protokolle konnten sich in der Praxis aber nicht durchsetzen oder wurden von POP3 verdrängt.

IMAP - Internet Message Access Protocol

Genau wie POP3 setzt IMAP auf TCP als Basisprotokoll auf und dient dazu, empfangene E-Mails in die Client-Anwendung zu transportieren. Im Gegensatz zu POP3 belässt IMAP empfangene E-Mails auf dem Server und holt nur eine Kopie der E-Mail zur Ansicht in die Client-Anwendung.

Für den Anwender hat das den Vorteil, dass ein E-Mail-Konto von verschiedenen Endgeräten wie PC, Notebook, Smartphone oder Tablet genutzt werden kann und alle Geräte den gleichen Empfangsstand sehen.

Eine weitere Neuerung von IMAP ist die Möglichkeit, auf dem Mailserver die empfangenen E-Mails in Ordnern abzulegen und zu verwalten.

E-Mail per SMTP mit Authentisierung

SMTP in seiner ursprünglichen Form sieht nicht vor, dass der Benutzer, der E-Mails versenden möchte, sich in irgendeiner Form authentisieren, also seine Berechtigung nachweisen muss.

Das bedeutet: Jeder, der Zugang zu dem Netzwerk hat, in welchem der SMTP-Server platziert ist, kann von dort aus E-Mails versenden.

Im Zeitalter von Internet, Spam (unerwünschte Werbe-E-Mail) und Computerviren ist das natürlich ein nicht tragbarer Zustand.

Deshalb wurden Authentisierungsverfahren entwickelt, die nur dem berechtigten Benutzer erlauben, E-Mails über den Server zu verschicken.

Die zwei gängigsten Verfahren möchten wir hier kurz vorstellen.

SMTP after POP3

Diese Methode ist denkbar einfach. Nur solche User, die auf dem Mailserver ein POP3-Postfach haben, sind berechtigt, über diesen Server E-Mails zu versenden.

Bevor das Senden von E-Mails zugelassen wird, muss ein Login in das POP3-Postfach erfolgen.

Der Vorteil dieser Methode ist, dass jedes normale Mailprogramm nach dem Start zunächst das POP3-Postfach nach neu eingegangenen E-Mails durchsucht und über den damit verbundenen POP3-Login automatisch die Voraussetzung zum Versenden von E-Mails schafft.

Der Anwender muss also keine besondere Konfiguration an seinem Mailprogramm vornehmen.

Nur bei Embedded Endgeräten wie z.B. Web-IOs oder Web-Thermometern sollte darauf geachtet werden, dass bei SMTP Authentification „SMTP after POP3“ eingestellt wird, da diese keine E-Mails empfangen und deshalb nicht automatisch auf das POP3-Postfach zugegriffen wird.

ESMTP - Extended SMTP

Wird ESMTP benutzt, erfolgt die Authentisierung innerhalb der SMTP-Kommunikation. E-Mails können unabhängig vom POP3-Zugang versendet werden.

Nachdem die TCP-Verbindung zum SMTP-Server zustande gekommen ist, fragt dieser zunächst nach einem Usernamen und dem zugehörigen Passwort.

Erst wenn beides richtig übergeben wurde, können E-Mails versendet werden.

Für den Betrieb von Embedded Geräten hat diese Methode den Vorteil, dass zum

Versenden von E-Mails nur eine TCP-Verbindung nötig ist.

Normale Mailprogramme müssen für den ESMTP-Betrieb speziell konfiguriert werden.

Verschlüsselter E-Mailversand mit SSL/TLS
Vor allem bei den öffentlichen E-Mail-Providern wird inzwischen erwartet, dass der Datenaustausch beim E-Mailversand verschlüsselt erfolgt.

Für den Anwender ändert sich dadurch eigentlich nicht viel. Die Verschlüsselung erledigen Mailclient und Mailserver sozusagen unter der Haube und für den Anwender gar nicht wahrnehmbar.

Das dazu verwendete SSL/TLS-Verfahren wird im Kapitel Datensicherheit/Netzwerksicherheit näher erläutert.

Der Anwender muss allerdings berücksichtigen, dass bei verschlüsselter E-Mailübertragung andere TCP-Ports genutzt werden, die je nach Provider auch variieren können:

- SSL/TLS SMTP: TCP-Port 465 (oder auch 587)
- SSL/TLS POP3: TCP-Port 995
- SSL/TLS IMAP: TCP-Port 993

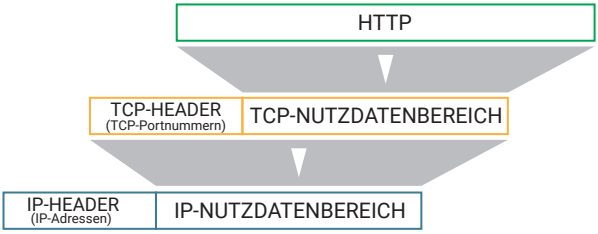
E-Mail über HTTP senden und empfangen

Mit der zunehmenden Nutzung von E-Mail haben sich immer mehr Freemail-Anbieter etabliert, die auf ihrem Mailserver kostenlos Postfächer zur Verfügung stellen. Diese Dienstleistung, die jeder nutzen kann, wird in aller Regel über Werbung finanziert.

Um Raum zur Einblendung von Werbung zu schaffen, geben die meisten Freemail-Anbieter dem Nutzer die Möglichkeit, das Senden und Abrufen von E-Mails bequem über HTTP im Browser abzuwickeln, der selbstverständlich durch Werbebanner bereichert ist. Hierzu stehen dem Nutzer entsprechende HTML-Formulare zur Verfügung.

Um die E-Mail-Abwicklung über HTTP zu ermöglichen, muss der Freemail-Anbieter eine spezielle Mailserver-Kombination betreiben, die zur Nutzerseite als Webserver und zur anderen Seite als SMTP-Server arbeitet. Der Weg einer E-Mail sieht hier folgendermaßen aus:

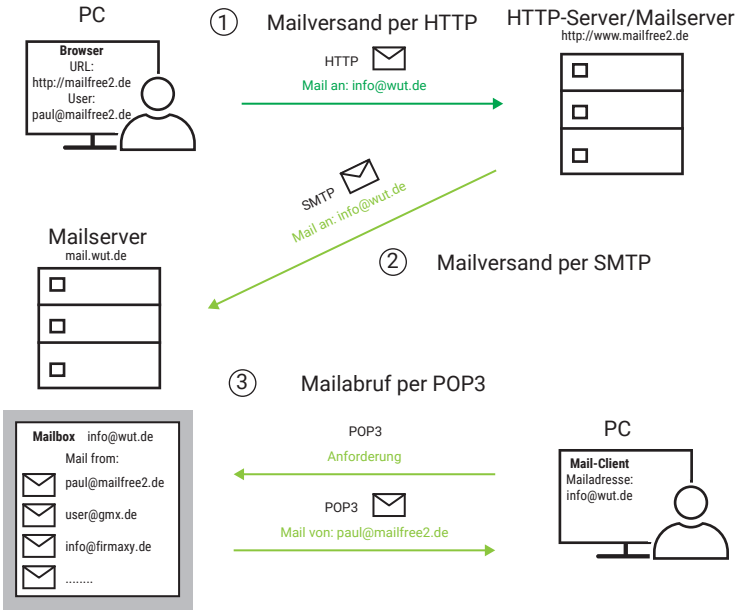
1. Zwischen dem Rechner des Absenders und dem Server des Freemail-Anbieters wird das HTTP-Protokoll verwendet. Wie bei anderen HTTP-Anwendungen wird auch hier die TCP-Portnummer 80 genutzt.



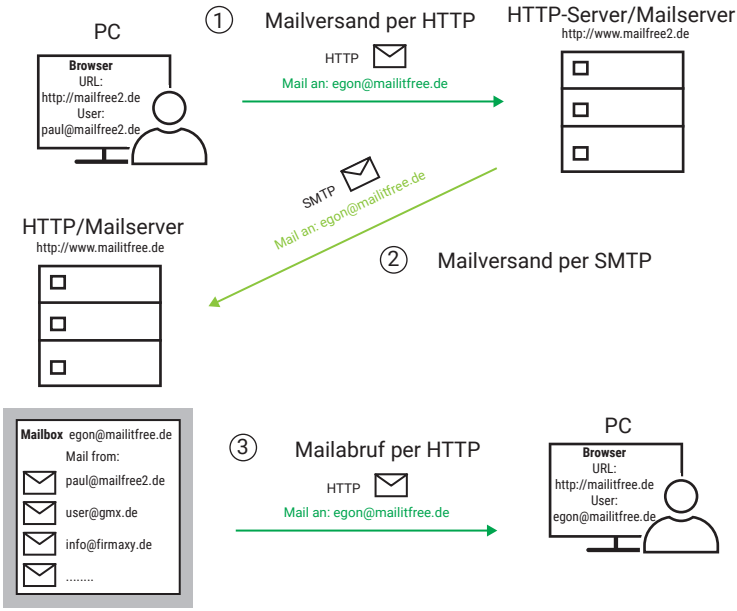
2. Zwischen den Mailservern selbst ändert sich nichts. Sie kommunizieren miteinander über das SMTP-Protokoll.

3. Zwischen dem Ziel-Mailserver und dem Rechner des Empfängers können zwei unterschiedliche Varianten zum Einsatz kommen:

   Hat der Empfänger ein Standard-Mailkonto, werden eingegangene Mails über POP3 abgeholt.



Nutzt auch der Empfänger die Dienste eines Freemail-Anbieters, kommt hier ebenfalls HTTP zum Einsatz.



Wer seine E-Mail lieber über SMTP und POP3 versenden möchte, sollte bei der Wahl des Freemail-Anbieters unbedingt darauf achten, dass auch Zugangsmöglichkeiten über einen SMTP- bzw. POP3-Server vorhanden sind.

E-Mail und DNS

Auch beim Versenden von E-Mails wird auf IP-Ebene mit IP-Adressen gearbeitet. Die Namensauflösung bei E-Mail-Adressen funktioniert im Prinzip genauso wie bei normalen Netzteilnehmern auch. Natürlich wird dabei nicht die Adresse des E-Mail-empfängers selbst aufgelöst, sondern lediglich die des Mailservers, auf dem der Empfänger sein Postfach hat.

Zur Erinnerung: Um Namen in Adressen aufzulösen, bedient sich der TCP/IP-Stack eines Resolver-Programms, das beim DNS-Server eine entsprechende Anfrage stellt.

Nun ist der Hostname des Ziel-Mailservers aber nicht bekannt. Bekannt ist lediglich die Ziel-Domain, die ja in der E-Mail-Adresse hinter dem @-Zeichen steht. Um auch

DNS-Anfragen nach Mailservern auflösen zu können, gibt es auf DNS-Servern spezielle Datensätze, in denen die zu einer Domain gehörenden Mailserver samt der zugehörigen IP-Adressen verzeichnet sind.

Das Resolver-Programm gibt also bei der Anfrage nur den Ziel-Domainnamen an und teilt zudem mit, dass es sich bei dem gesuchten Netzteilnehmer um einen Mailserver handelt. Der DNS-Server ermittelt die gesuchte IP-Adresse und gibt sie an das Resolver-Programm zurück.

Der Postfachname selbst wird für die DNS-Anfrage nicht benötigt. Er wird erst bei Eintreffen der Nachricht auf dem Ziel-Mailserver ausgewertet, damit diese im richtigen Postfach abgelegt werden kann.

Industrieprotokolle bis IoT

Nicht erst seit dem Hype um das Thema Industrie 4.0 gibt es Bestrebungen, standardisierte Kommunikationsmöglichkeiten für den industriellen Einsatz zu schaffen.

In der Vergangenheit wurde dabei häufig auf Feldbussysteme gesetzt, also serielle Verbindungen zwischen den beteiligten Komponenten. Dabei haben sich diverse Standards nebeneinander etabliert, die sich nicht nur in Protokoll und Übertragungsgeschwindigkeit unterscheiden. Auch die physikalische Übertragung bis hin zu den verwendeten mechanischen Anschlussmöglichkeiten variieren stark.

Die im Folgenden vorgestellten Industrieprotokolle unterscheiden sich zwar auf Protokollebene, aber als physikalisches Transportmedium nutzen alle TCP/IP-Ethernet.

Damit gibt es einen gemeinsamen Standard, der viele Vorteile mit sich bringt:

- vorhandene Infrastruktur kann genutzt werden
- unterschiedliche Industrieprotokolle können im gleichen Netzwerk nebeneinander verwendet werden
- einheitliche Übertragungstechnik und Steckverbinder
- standortübergreifende Kommunikation möglich
- beliebig erweiterbar

IoT und Industrie 4.0

Beide Begriffe sind zur Zeit in aller Munde, werden aber zum Teil recht unterschiedlich interpretiert. Deshalb beschränken wir uns hier auf eine kurze Beschreibung und konzentrieren uns danach auf die verwendeten Protokolle.

IoT - Internet of Things

Die rasante Ausbreitung des Internets ist sicher eine der größten Errungenschaften unserer Zeit. Bis vor kurzem bestand die Hauptnutzung des Internets darin, dass ein Nutzer, also ein Mensch am PC oder Smartphone, im Browser Webseiten abgerufen oder E-Mails verschickt hat. Dienste wie Twitter, WhatsApp und diverse Smartphone-Apps kamen dazu. Ebenso das Streamen von Videos.

Aber immer war es so, dass ein Mensch an diesem Informationsaustausch beteiligt war.

Beim Internet of Things (Internet der Dinge) geht es darum, nicht nur PCs, Tablets und Smartphones den Zugang zum Internet zu ermöglichen. Letztlich sollen Geräte mit verschiedensten Funktionen über das Internet miteinander kommunizieren.

Die Anwendungsfelder reichen von der simplen Temperaturüberwachung über SmartHome bis hin zum autonomen Fahren von Autos. Auch in der Industrie wird das Internet zunehmend zum Kommunikationsweg für den Austausch von Maschinen- und Fertigungsdaten.

Industrie 4.0

Der Begriff Industrie 4.0 mutet schon etwas merkwürdig an, weil man vorher nie etwas von Industrie 1.0, 2.0 oder 3.0 gehört hatte. Letztlich wurde der Begriff von der Politik ins Spiel gebracht, um die Bedeutung von Digitalisierung im industriellen Umfeld hervorzuheben.

In diesem Zuge wurden auch die vorangegangenen Generationen bestimmten Epochen zugeordnet.

- **Industrie 1.0**
  Ab 1800 wurden zunehmend Maschinen zur Massenproduktion eingesetzt. Mit Erfindung der Dampfmaschine konnten Maschinen, wie z.B. Webstühle, auch unabhängig von der bis dahin genutzten Wasserkraft betrieben werden. Immer mehr Fabriken entstanden.

- **Industrie 2.0**
  Mit der Einführung der Elektrizität wurden Ende des 19. Jahrhunderts die Antriebe für Maschinen kleiner und leichter. Fließband und Akkordarbeit hielten Einzug in die Fabriken.

- **Industrie 3.0**
  Ab den 1970er Jahren wurden die ersten Computer in Fertigungsanlagen eingesetzt. SPS-Steuerungen (Speicherprogrammierbare Steuerungen) überwachen, steuern und automatisieren Produktionsprozesse.

- **Industrie 4.0**
  Das von der Politik angedachte Ziel von Industrie 4.0 ist die individuelle und autarke Massenfertigung. Die Vorstellung geht dahin, dass alle an einem Fertigungsprozess beteiligten Komponenten den Prozess automatisiert mitgestalten.

> Ziel ist dabei die „Losgröße 1“, was soviel bedeutet, dass moderne Fertigung so flexibel arbeiten sollen, dass zwischen einer Massenfertigung auch Einzelstücke produziert werden können. Und zwar ohne Eingriff von außen und ohne Maschinenumrüstung.

Insbesondere die Grenze zwischen Industrie 3.0 und Industrie 4.0 ist fließend. Mit immer leistungsfähigerer Hardware wächst der Bedarf nach Datenaustausch stetig. Hinsichtlich der Datenkommunikation gibt es schon lange die einheitliche Nutzung von TCP/IP sowohl im Nahbereich innerhalb der Firmennetze, wie auch im Fernbereich unter Nutzung des öffentlichen Internets. Neu ist in diesem Kontext der zunehmende Sicherheitsanspruch. Mehr dazu erfahren Sie im Kapitel *Datensicherheit/ Netzwerksicherheit*.

Fakt ist, dass die zunehmende Digitalisierung in den Werkhallen bereits im vollen Gange und nicht mehr aufzuhalten ist.

Die wichtigsten etablierten Standards der industriellen TCP/IP-basierten Datenkommunikation werden wir im Folgenden kurz vorstellen.

Nachrichtenformate

In den vorangegangenen Kapiteln haben wir bereits einige Protokolle kennengelernt. Eines haben alle Protokolle gemeinsam: Es gibt grundsätzliche Adressinformationen und die eigentlichen Daten, die übertragen werden sollen. Optional gibt es Checksummen oder andere Informationen zur Sicherung der Daten.

In welcher Form die transportierten Daten übertragen werden, hängt von der Anwendung ab und ist bei den meisten Protokollen ebenfalls vorgegeben.

Es gibt zwei grundsätzliche Datenformate:

- Binäre Daten
- Nachrichtentext

Wann welche Variante zum Tragen kommt, hängt von vielen Faktoren ab.

Binärdaten

Zur Erinnerung: Daten sind immer eine bestimmte Anzahl an Bytes.

Welches Byte an welcher Stelle welchen Zweck erfüllt, ist entweder durch ein stan-

dardisiertes Protokoll oder die Anwendung festgelegt. Hinter einem oder mehreren Bytes verbirgt sich ein Wert, ein Werte-Array, eine Zeichenkette oder auch ein Funktionsaufruf.

Dabei können in einer Datensendung einzelne Werte übertragen werden. Oft wird aber auch mit Datenstrukturen gearbeitet, bei denen festgelegt ist, an welcher Stelle der übertragenen Byte-Kette welcher Wert abgelegt ist.

Hier als Beispiel Daten eines Modbus-Funktionsaufrufs. Der Function Code ist z.B. immer im 8. Byte untergebracht:

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

Ein weiteres gängiges Verfahren für binären Datenaufbau ist TLV, was für Type Length Value steht. Mehrere Inhalte beliebiger Größe können aufeinanderfolgend in einer Datensendung übertragen werden.

Für jeden Inhalt gilt die Abfolge:

1. Type - um was für eine Art von Inhalt handelt es sich?
   Typenfestlegung bestimmt durch die Anwendung
2. Length - wie viele Bytes umfasst der Inhalt?
3. Value - Bytes des Wertes bzw. Inhalts.

Gibt es hinter einer solchen Abfolge weitere Bytes, ist das die nächste Abfolge.

Hier ein einfaches Beispiel:

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

Die übermittelten Bytes beinhalten zwei Werte: einen 16Bit-Wert (2 Bytes) und einen 32Bit-Wert (4 Bytes).

Der Vorteil der binären Datenübertragung ist der sehr kompakte Aufbau der Daten.

Daten als Text

Insbesondere bei webbasierten Anwendungen werden Daten aller Art als Text versendet. Text bedeutet, dass die Informationen als für den Menschen lesbare Zeichenkette übertragen werden. Dabei belegt jedes Zeichen ein Byte.

Die Kodierung erfolgte in der Vergangenheit nach dem ASCII-Standard. Die Zuordnung, welches Schriftzeichen welchem Zahlenwert entspricht, ist in der ASCII-Tabelle definiert (ASCII = American Standard Code for Information Interchange).

Die Besonderheit dabei war in der Vergangenheit, dass nur 7 der 8 zur Verfügung stehenden Bits eines Bytes genutzt wurden, womit der verwendbare Zeichenvorat auf 128 lesbare Zeichen beschränkt ist.

Neuere Standards wie UTF8 überwinden diese Einschränkung und erlauben es für Sonderzeichen, sogar zwei Bytes für ein Zeichen zu verwenden.

Neben frei formulierten Textinhalten haben sich standardisierte Textformate bei den Web- und Industrieprotokollen etabliert:

- XML
- JSON

Beide Formate wollen wir hier kurz erklären.

XML- Extensible Markup Language

XML ist eine sogenannte Auszeichnungssprache. Die eigentlichen Nutzdaten werden in Tags eingebettet. Die Tags sind Benennungen der jeweiligen Werte bzw. Inhalte. Jedes Tag beginnt mit einer öffnenden spitzen Klammer und endet mit einer schließenden.

Jedes XML-Konstrukt beginnt mit einem Start-Tag, in dem mindestens die XML-Version angegeben ist. Zusätzliche Parameter, wie beispielsweise die verwendete Zeichenkodierung, sind ebenfalls möglich:

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

Nach dem Start-Tag folgen die weiteren in Tags eingebetteten Inhalte. Alle Inhalte bis auf das Start-Tag haben ein öffnendes und ein schließendes Tag gleichen Namens. Allerdings beginnt die Benennung beim schließenden Tag mit einem Slash ("/").

Beispiel:

```
<inhalt>irgendetwas</inhalt>
```

XML lässt auch strukturiert, nach Hierarchie ineinander verschachtelte Tags zu. Hier als Beispiel die Sensorwerte eines W&T Web-Thermo-Hygrobarometers:

```
<?xml version="1.0" encoding="UTF-8"?>
<webio>
  <iostate>
    <sensor>
      <name>Temperatur</name>
      <number>0</number>
      <unit>°C</unit>
      <value>23.900000</value>
    </sensor>
    <sensor>
      <name>rel. Feuchte</name>
      <number>1</number>
      <unit>%</unit>
      <value>36</value>
    </sensor>
    <sensor>
      <name>Luftdruck</name>
      <number>2</number>
      <unit>hPa</unit>
      <value>992</value>
    </sensor>
  </iostate>
</webio>
```

Die Einrückungen sind bei XML keine Pflicht, aber üblich, da die Lesbarkeit dadurch deutlich erhöht wird.

Der Vorteil bei XML als Übertragungsformat liegt darin, dass sowohl Mensch, als auch Maschine bzw. ein auswertendes Programm die Inhalte gut lesen können.

Nachteil ist das sehr hohe Brutto-Datenaufkommen für wenige Inhalte.

JSON - JavaScript Object Notation
Die Syntax, also der Aufbau von JSON, basiert auf einer Teilmenge der JavaScript-Syntax.

JSON nutzt Paare aus Name und Wert/Inhalt zur Kodierung der Daten.

Beispiel: "inhalt" : "irgendetwas"

Auch JSON lässt einen strukturierten, nach Hierarchie ineinander verschachtelten Aufbau zu. Hier als Beispiel, noch einmal die Sensorwerte eines W&T Web-Thermo-Hygrobarometers:

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

Sowohl Namen als auch Werte werden in Anführungszeichen oben eingebettet. Eine Ausnahme sind Zahlenwerte - hier kann auf die Anführungszeichen verzichtet werden.

Name/Werte-Paare sind durch Kommata getrennt.

Zusammengehörende Name-/Werte-Paare müssen mit geschweiften Klammern zu Gruppen zusammengefasst werden.

Zusammengehörende Gruppen können ein Array bilden und werden durch Kommata getrennt in eckigen Klammern zusammengefasst.

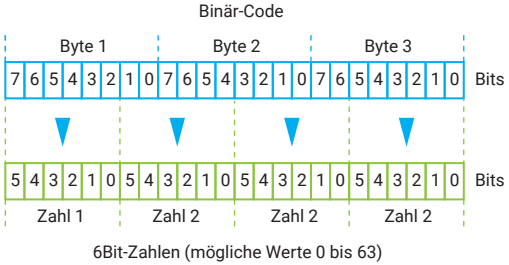
Eine detaillierte Beschreibung zum JSON-Format gibt es unter *https://www.json.org*.

JSON ist vom Datenaufkommen deutlich kompakter als XML und trotzdem von Mensch und Maschine gut lesbar.

Base64-Kodierung

Base64 ist ein Verfahren, das Binärdaten in eine Kette lesbarer ASCII-Zeichen kodiert bzw. dekodiert. Auf diese Weise lassen sich binäre Inhalte auch mit Text basierenden Übertragungsformaten transportieren.

Das Verfahren ist recht einfach. Es werden jeweils drei Bytes des Binär-Codes bitweise auf vier 6Bit-Zahlen übertragen.



Jeder der vier Zahlen wird nach der folgenden Tabelle das dem Wert entsprechende Zeichen zugeordnet. So werden drei binäre Bytes durch vier Chars, also lesbare Zeichen, ersetzt.

Wert		Char	Wert		Char	Wert		Char	Wert		Char
dez.	hex.		dez.	hex.		dez.	hex.		dez.	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	/

Dieser Vorgang wird wiederholt, bis die gesamten binären Bytes kodiert sind. Blei-

ben am Ende einzelne Bytes über, werden Füllbytes hinzugefügt, um die letzten drei Bytes zu kodieren. Füllbytes haben den Wert 0.

Um beim anschließenden Dekodieren, also der Rückgewinnung der ursprünglichen Binärbytes, die Füllbytes wieder aussortieren zu können, wird der kodierten Zeichenkette für jedes Füllbyte am Ende ein "="-Zeichen angehängt.

Die häufigsten Anwendungsfälle für Base64-Kodierung sind webbasierte Anwendungen und E-Mail.

Modbus-TCP

Ursprünglich wurde Modbus als serieller Feldbus von der Firma Modicon (heute Schneider Electric) als Kommunikationsweg zwischen deren Steuerungen entwickelt.

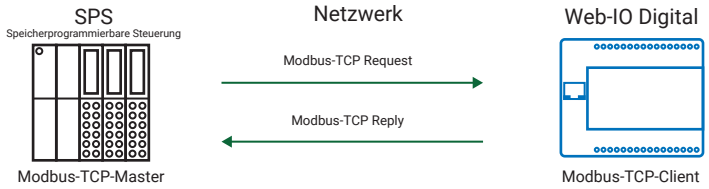
Der klare und einfache Aufbau des Modbus-Protokolls hat dazu geführt, dass auch andere Hersteller Modbus in ihre Geräte integriert haben. Damit hat Modbus sich zu einem bis heute etablierten Standard entwickelt.

Das Master/Slave-Prinzip

Modbus arbeitet nach dem Master/Slave-Prinzip. Das bedeutet, dass es mindestens einen Master und mindestens einen Slave gibt.

Modbus-Slaves sind z.B. SPS-Steuerungen, Web-IOs oder andere dezentrale IO-Baugruppen für digitale und analoge Signale.

Der Master ist immer der Kommunikationspartner, der die Initiative ergreift, also eine Anfrage bzw. den gewünschten Funktionsaufruf an einen Slave sendet. Jeder Slave hat eine eindeutige Adresse. Der Slave ist im Normalfall rein passiv und antwortet nur dann, wenn er gezielt mit seiner Adresse angesprochen wird.



Mit der zunehmenden Bedeutung von TCP/IP-Ethernet als Übertragungsmöglichkeit wurde das Modbus-Protokoll nahezu 1:1 von serieller Datenübertragung auf TCP adaptiert.

Modbus-TCP arbeitet nach dem Client/Server-Prinzip, wobei der Master den Part des Clients übernimmt und die Slaves als Server agieren. Der Modbus-Master muss also zu jedem Modbus-Slave eine explizite TCP-Verbindung aufbauen. Diese Verbindung bleibt für die Dauer der Kommunikation bestehen. Es wird nicht für jede Anfrage eine neue Verbindung aufgebaut.

Der standardisierte TCP-Server-Port für Modbus-TCP ist 502.

Der Informationsaustausch erfolgt über das Lesen oder Beschreiben von Speicheradressen durch den Modbus-Master, also den Client.

Man kann sich das so vorstellen, als hätte der Modbus-Slave, also der Server, einen Schrank mit sehr vielen Schubfächern, die alle durchnummeriert sind. Den Schubfächern sind Funktionen zugeteilt.

Will der Modbus-Master bestimmte Informationen abrufen, gibt er in seiner Anfrage die Nummer des entsprechenden Fachs an und bekommt vom Modbus-Slave den Inhalt zurück.

Wenn der Modbus-Master beim Slave etwas auslösen möchte, z.B. einen Schaltausgang bedienen, legt er die notwendige Information in das Fach mit der entsprechenden Nummer.

Wie eingangs beschrieben, erfolgt das tatsächlich über entsprechende Speicheradressen. Es stehen maximal 65536 Adressen zur Verfügung. Welche Funktion sich hinter welcher Speicheradresse verbirgt, wird vom Gerätehersteller festgelegt - ist also nicht einheitlich vorgegeben.

In aller Regel ist der Speicher getrennt nach Funktionen in Bereiche aufgeteilt.

Es gibt vier verschiedene Datentypen, auf die zugegriffen werden kann:

Bezeichnung	Daten Typ	Zugriff	Beschreibung
Discrete Input	1-Bit	nur lesen	digitaler Input bzw. Schaltzustand

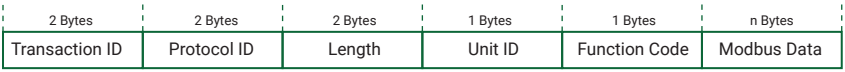
Bezeichnung	Daten Typ	Zugriff	Beschreibung
Coil	1-Bit	lesen/ schreiben	digitaler Output bzw. Schaltzustand
Input Register	16-Bit	nur lesen	Wert zwischen 0 und 65535 bzw. Analogwert oder Zählwert
Holding Register	16-Bit	lesen/ schreiben	Wert zwischen 0 und 65535 bzw. Analogwert oder Zählwert

Über Function Codes wird innerhalb des Modbus-Protokolls angegeben, auf welchen Datentyp wie zugegriffen werden soll. Hier eine Liste der gängigsten Funktionscodes:

FC (dez.)	FC (hex.)	Beschreibung
01	0x01	Read Coils Lesen digitaler Outputs bzw. Schaltausgänge
02	0x02	Read Discrete Inputs Lesen digitaler Inputs bzw. Schaltzustände
03	0x03	Read Holding Registers Lesen von 16-Bit (Output-)Registern
04	0x04	Read Input Registers Lesen von 16-Bit (Input-)Registern
05	0x05	Write Single Coil Schreiben eines einzelnen Outputs bzw. Schaltausg.
06	0x06	Write Single Register Schreiben eines einzelnen 16-Bit Registers
15	0x0F	Write Multple Coils Schreiben mehrerer Outputs bzw. Schaltausgänge
16	0x10	Write Multiple Registers Schreiben mehrerer 16-Bit (Output-)Register
07	0x07	Read Exeption Status Fehlerzustand anfordern

Modbus-TCP Protokollaufbau

Der Modbus-TCP-Protokollrahmen hat folgenden Aufbau:



Transaction ID
Die Transaction ID ist so etwas wie eine Anfragenummer und wird vom Master bei jeder Anfrage um eins hochgezählt. Der Client antwortet mit der gleichen Transaction ID.

Protocol ID
Bei Modbus-TCP immer 0.

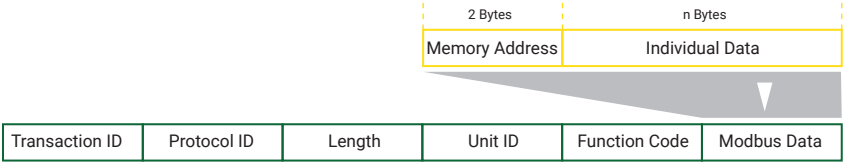
Length
Länge der Modbus-Daten in Byte plus zwei.

Unit ID
Beim seriellen Modbus-Protokoll war das die Adresse des Slaves. Das Feld ist aus Kompatibilitätsgründen übernommen worden. Bei Modbus-TCP-erfolgt die eindeutige Adressierung allerdings über die IP-Adresse des Slaves.

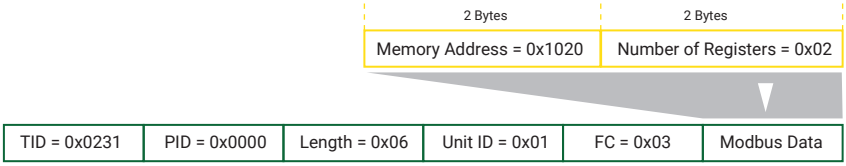
Function Code
Das Modbus-Protokoll definiert über durchnummerierte Function Codes, was die vom Master gesendete Anfrage beim Slave auslösen soll.

Modbus Data
Der Modbus Data Bereich wird abhängig vom verwendeten Function Code mit unterschiedlichen Inhalten gefüllt und kann dadurch unterschiedlich groß ausfallen. Auch die Datenrichtung spielt beim Aufbau des Modbus Data Bereichs eine Rolle.

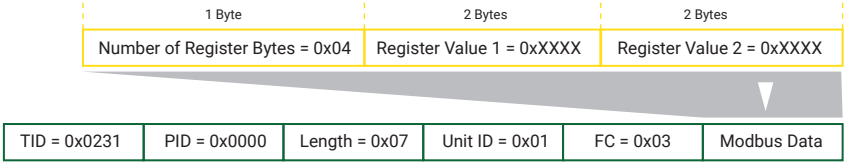
Bei Datenrichtung Master an Slave beinhalten die ersten zwei Bytes immer die anzusprechende Speicheradresse.



Das folgende Beispiel zeigt, wie ein Modbus-TCP Paket beim Abrufen von zwei Registern mit Function Code 3 ab Speicheradresse 0x1020 aussieht.



Das Antwortpaket ist anders aufgebaut. Hier ist im ersten Byte von Modbus Data die Anzahl der übergebenen Register-Bytes kodiert. In den nächsten 4 Bytes sind die Inhalte der angeforderten Register.



Trotz des überschaubaren Protokollaufbaus bietet Modbus-TCP große Flexibilität für die industrielle Kommunikation.

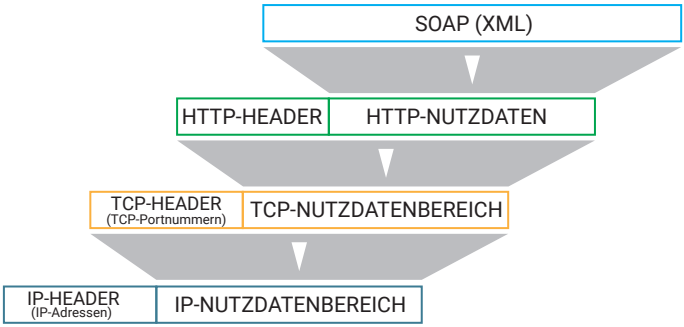
Tipp: *Achten Sie darauf, ob der Gerätehersteller bei der Beschreibung der Speicheradressen die Werte dezimal oder hexadezimal angibt.*

SOAP - Simple Object Access Protocol

SOAP ist ein webbasiertes Nachrichtenprotokoll, das nicht nur im industriellen Umfeld eingesetzt wird.

Übertragung auf Netzwerkebene

SOAP ist nicht an ein spezielles Transportprotokoll gebunden. Als webbasierendes Protokoll setzt SOAP aber im Allgemeinen auf HTTP bzw. HTTPS auf. Es gibt aber auch seltene Anwendungen, bei denen z.B. FTP oder SMTP als Transportprotokoll genutzt wird. Wir beschränken uns hier auf die Beschreibung von SOAP in Verbindung mit HTTP(S).



Der Vorteil von HTTP(S) als Kommunikationsgrundlage liegt darin, dass die meisten Netzwerke, auch wenn sie durch Firewalls usw. geschützt sind, für HTTP(s) durchgängig nutzbar sind.

Genutzt werden, wie bei HTTP bzw. HTTPS üblich, die TCP-Ports 80 bzw. 443.

Bedingt durch die Nutzung von HTTP(S) arbeitet SOAP nach dem Client/Server-Prinzip. Der Kommunikationsablauf ist immer gleich. Der Client schickt einen HTTP-Request als POST-Methode zum Server. Mit dem Post übergibt der Client die entsprechenden Daten im XML-Format.

Der Server verarbeitet die übergebenen Daten und sendet eine entsprechende Bestätigung.

Das Nachrichtenformat

Als Nachrichtenformat nutzt SOAP die XML-Syntax. Alle Informationen sind in XML-Tags eingebettet. Dabei folgen SOAP-Nachrichten einem vorgegebenen, strukturierten Aufbau.

Hinter der Angabe der XML-Version beginnt der eigentliche SOAP-Teil, der von Envelope-Tags umschlossen ist. Das Envelope-Tag beinhaltet als Parameter einen Verweis darauf, dass es sich um das standardisierte SOAP-Format handelt und dass die Inhalte entsprechend kodiert sind.

Eingebettet in die Envelope-Tags sind ein Nachrichtenkopf und die eigentlichen Daten.

Der Nachrichtenkopf ist optional und wiederum von Header-Tags umschlossen.

Wenn er vorhanden ist, beinhaltet er Angaben zum Umgang mit den eigentlichen Daten.

Die Daten selbst sind von Body-Tags umschlossen.

```
<?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>
```

Wie die Daten innerhalb des Body-Tags strukturiert sind und welche Tags genutzt werden, gibt die Anwendung vor.

Vorteile von SOAP
- lesbarer Text
- von vielen Software-Herstellern unterstützt
- unabhängig von Betriebssystem und Programmiersprache

Nachteile von SOAP
- Hoher Overhead, also hohes Datenaufkommen, da alle übertragenen Daten in XML-Tags eingeschachtelt sind.
- Binäre Inhalte müssen zunächst mittels MIME- oder Base64-Kodierung in darstellbaren Text und später zurückgewandelt werden.

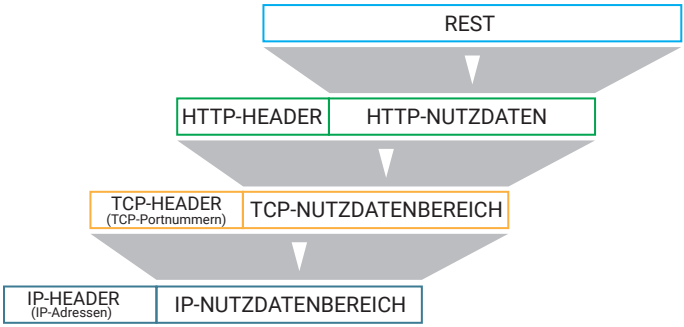
Weitere Details zum SOAP-Standard finden Sie unter: https://www.w3.org/TR/soap/

REST - REpresentational State Transfer

REST beschreibt keinen konkreten Protokollaufbau. Stattdessen gibt REST Eigenschaften vor, die für einen Datenaustausch im industriellen Umfeld erfüllt sein sollten.

Übertragung auf Netzwerkebene

Vorab sei gesagt, dass REST als übergeordnetes Protokoll HTTP bzw. HTTPS nutzt.



Der Vorteil von HTTP(S) als Kommunikationsgrundlage liegt darin, dass die meisten Netzwerke, auch wenn sie durch Firewalls usw. geschützt sind, für HTTP(s) durchgängig nutzbar sind. Außerdem wird HTTP(S) den für REST geforderten Eigenschaften gerecht.

Genutzt werden, wie bei HTTP bzw. HTTPS üblich, die TCP-Ports 80 bzw. 443.

Eigenschaften und Grundelemente

Client/Server-Modell

REST arbeitet nach dem Client/Server-Verfahren, wobei die Aktion immer vom Client angestoßen wird. Der Server stellt Ressourcen (Daten, Inhalte, Funktionen) zur Verfügung. Der Client sendet einen Request, um auf ausgewählte Ressourcen zuzugreifen. Der Server liefert mit einem Reply die angeforderten Daten oder bestätigt die gewünschte Aktion.

Zustandslosigkeit

Bei vielen Client/Server-Anwendungen wird beim Verbindungsaufbau serverseitig ein bestimmter Status (Berechtigung, spezielle Aufgabe, spezifischer Zweck ….) zugewiesen, der für die Dauer der Verbindung erhalten bleibt. REST behandelt alle Sendungen zum Server zunächst gleich und erst der Inhalt der Datensendung bestimmt den weiteren serverseitigen Umgang bzw. die Einordnung des Requests. Der Status einer Anwendung liegt also in den Händen des Clients.

Geschichtetes System

REST sieht eine klare Trennung von Zuständigkeiten vor. Diese Trennung gilt insbesondere für die Abwicklung der Kommunikation und die Weiterverarbeitung der

transportierten Inhalte.

Dadurch ist es möglich, Proxies, Gateways oder ähnliche Zwischenstationen auf dem Übertragungsweg zu nutzen.

Außerdem kann je nach Sicherheitsanspruch HTTP oder die verschlüsselte Übertragung per HTTPS für die Kommunikation gewählt werden. Für die funktionale Abwicklung der REST-Inhalte macht das keinen Unterschied.

Adressierbarkeit der Ressourcen
Alle auf einem Server verfügbaren Ressourcen sind über eine eindeutige Adresse (URI = Uniform Resource Identifier) abrufbar. Der URI ist aufgebaut, wie die URL, die für die Adressierung im Browser verwendet wird:

```
Protokoll://<Host>:<Port>/<Pfad>/<Ressource>?<Parameter>&<Parameter>
```

Anfrage-Methoden
Für die Requests stehen die standardisierten HTTP-Aufrufe zur Verfügung:

- GET ruft Ressourcen vom Server ab und wird ausschließlich lesend genutzt
- POST legt neue Ressourcen an oder ändert bestehende
- PUT ändert bestehende Ressourcen
- DELETE löscht bestehende Ressourcen
- HEAD fordert nur den HTTP-Kopf an, um z.B. die Verfügbarkeit der abzurufenden Ressource zu prüfen
- OPTIONS ruft Informationen zu den Kommunikationsoptionen ab

Bei den abgerufenen Daten spricht man von Repräsentationen einer oder mehrerer Ressourcen. Daher die Bezeichnung REST für REpresentational State Transfer.

Letztlich sind Repräsentationen nichts anderes als ein Abbild von Teilen der Prozessdaten. Die Übergabe kann in nahezu beliebiger, aber zu vereinbarender Form erfolgen. Üblich sind JSON, XML oder roher Text. Aber auch SVG, MP3 oder andere Formate werden abhängig von der Anwendung genutzt. Die Repräsentation kann auch Hyperlinks auf weitere Ressourcen beinhalten.

Im folgenden Beispiel sehen wir eine GET-Anfrage zur Abfrage der Sensorwerte an

ein Web-Thermo-Hygrobarometer von Wiesemann & Theis im JSON-Format:

Der Client sendet:

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

Und das Web-Thermo-Hygrobarometer sendet als Antwort:

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

Code on Demand

Bei Webseiten ist es heute üblich z.B. JavaScript mit- bzw. nachzuladen. Auch REST erlaubt, Quellcode, Programmteile oder Funktionsbausteine nachzuladen. Damit kann zur Laufzeit die Funktion des Clients erweitert oder verändert werden.

Cache

REST sieht vor, wiederholende Requests clientseitig aus einem Cache (Zwischenspeicher) zu beantworten, um die Datenlast auf den Übertragungswegen zu reduzieren. Der Server legt über entsprechende Angaben im HTTP-Kopf fest, ob für die angeforderte Repräsentation der Cache genutzt werden darf oder nicht.

Vorteile von REST

Einfache Implementierung, da weitestgehend die Mechanismen von HTTP genutzt werden

Nachteile von REST

Durch die Request-/Reply-Technik ist nur Pollen, also das gezielte Abrufen von Daten, aber keine eventgesteuerte Kommunikation möglich. Es müssen die relevanten Daten kontinuierlich gepollt (abgefragt) werden, um Veränderungen zu erkennen.

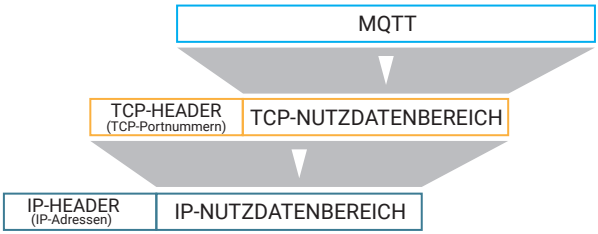
MQTT - Message Queue Telemetry Protocol

Die Besonderheit von MQTT besteht darin, dass zwei Kommunikationspartner, die miteinander Daten austauschen, das niemals über eine direkte Verbindung tun.

Stattdessen gibt es einen zentralen Datenvermittler, den Broker. Der Broker nimmt Daten von einem MQTT-Nutzer entgegen und verteilt sie an andere weiter.

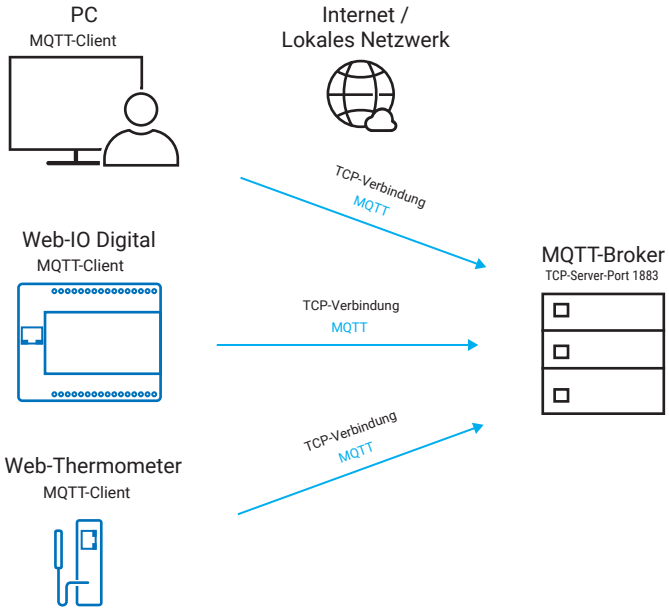
Übertragung auf Netzwerkebene

MQTT nutzt als Basisprotokoll TCP und arbeitet somit nach dem Client/Server-Prinzip.



Das MQTT-Datenpaket hat somit folgenden Aufbau:

Der Broker ist dabei der Server, der standardmäßig auf TCP-Port 1883 Verbindungen annehmen kann. Die MQTT-Nutzer agieren als TCP-Clients und verbinden sich bei Bedarf mit dem Broker.



Datenaustausch auf Protokollebene

Bei MQTT gibt es zwei Rollen, die ein Client annehmen kann.

Publisher

Als Publisher sendet der Client Daten an den Broker. Das können Messwerte, Schaltzustände oder andere beliebige Prozessdaten sein. Neben lesbaren Textinhalten lässt MQTT auch binäre Daten zu. Ob der Publisher seine Daten dem Broker bei Veränderung oder zyklisch übermittelt, hängt von der Anwendung ab.

Subscriber

Der Subscriber nimmt Daten vom Broker entgegen. In der Rolle des Subscribers teilt der Client dem Broker nach Verbindungsaufbau mit, welche Daten er erhalten bzw. abonnieren möchte.

Jeder MQTT-Client kann Publisher, Subscriber oder auch beides sein. Es gibt also kein zwingendes Master/Slave-Prinzip wie bei anderen Industrieprotokollen. Alle MQTT-Endgeräte bzw. Clients sind auf MQTT-Ebene zunächst gleichberechtigt. Wer Daten liefert und wer sie bekommt, bestimmt also ausschließlich die Anwendung durch die Publisher/Subscriber-Zuordnung.

Topic

Der MQTT-Broker verwaltet die auszutauschenden Daten nach Datenendpunkten. Die Benennung der Datenendpunkte erfolgt über Topics. Das sind Strings, also Zeichenketten, die ähnlich der URL beim Webseitenaufruf strukturiert aufgebaut sein können.

Beispiel:

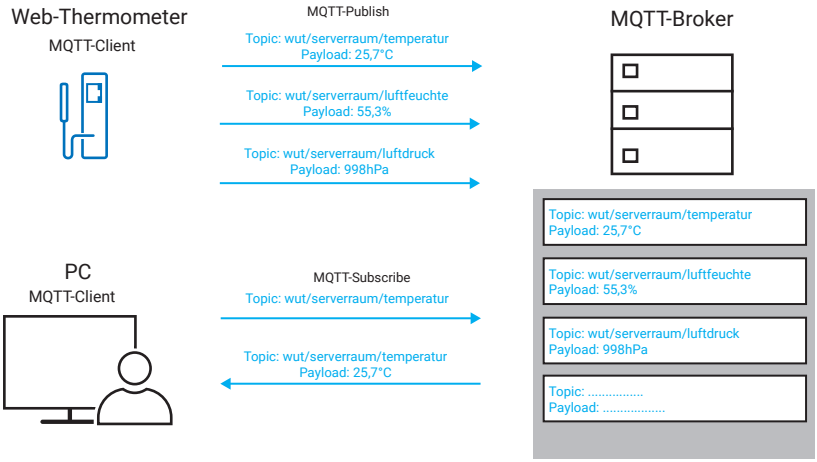
Ein W&T Web-Thermo-Hygrobarometer misst Temperatur, Luftfeuchte und Luftdruck im Serverraum des W&T Firmengebäudes. Jeder der drei Werte stellt einen Datenendpunkt dar.

Die Topics dazu könnten so aussehen:

```
wut/serverraum/temperatur
wut/serverraum/luftfeuchte
wut/serverraum/luftdruck
```

Per MQTT-Publish übergibt das Web-Thermo-Hygrobarometer die gemessenen Werte als Payload, also als Dateninhalt, unter diesen Topics an den MQTT-Broker.

Ein beliebiger MQTT-Client kann nun ein Subscribe unter Angabe des gewünschten Topics an den MQTT-Broker senden. Hier sendet der MQTT-Client ein Subscribe auf `wut/serverraum/temperatur` und abonniert damit den Temperaturwert des Web-Thermo-Hygrobarometers.



Solange der PC als MQTT-Client mit dem Broker verbunden ist, bekommt er automatisch den vom Web-Thermo-Hygrobarometer per Publish gesendeten Wert.

Der Subscriber kann bei der Angabe der Topics mit Wildcards „#“ arbeiten.

Beispiel: `wut/serverraum/#` abonniert Temperatur, Luftfeuchte und Luftdruck für den Serverraum.

Neben „#“ gibt es noch „+“ als Wildcard. Mit der Angabe von „+“ werden nur End-Topics der entsprechenden Ebene abonniert – Topics aus dahinterliegenden Ebenen werden ignoriert.

Für jeden abonnierten Wert gibt es eine eigene Datensendung.

Die Übertragung der Daten erfolgt byteorientiert und damit binärtransparent – es können also beliebige Inhalte übertragen werden. Für alle Textinhalte ist zur Vereinheitlichung UTF8 als Format vorgegeben.

Besondere Features

MQTT bietet für den Transport und Austausch der Daten einige Sonderoptionen, die über Flags für jede Verbindung bzw. jedes Abonnement gesetzt werden können.

Quality of Service – QoS

Mit Werten zwischen 0 - 2 wird festgelegt, mit welcher Zuverlässigkeit eine Publish-Nachricht an den Broker versendet wird.

0 raus und vergessen
Der MQTT-Client erwartet für gesendete Daten keine Bestätigung.
Dieses Verfahren ist unsicher, dafür aber sehr schnell.

1 mindestens einmal
Der MQTT-Client sendet die Daten ggf. auch mehrfach, bis er vom MQTT-Broker eine Empfangsbestätigung bekommt.
Dieses Verfahren stellt sicher, dass die Daten beim Broker ankommen, ggf. aber mehrfach.

2 genau einmal
Jede Datensendung muss explizit vom MQTT-Broker quittiert werden. Dadurch wird sichergestellt, dass nichts doppelt versendet wird.
Dieses Verfahren ist das sicherste, aber auch das langsamste.

Bei QoS1 und 2 stellt sich auf den ersten Blick die Frage, warum eine einfache Da-

tensendung sicherer ist als eine mehrfache. Anhand von zwei Beispielen wird der Hintergrund aber schnell klar.

Geht es darum, einen Messwert zu übertragen - z.B. eine Temperatur - spielt es keine Rolle, ob der Wert mehrfach bei einem Subscriber ankommt.

Wird aber der Winkel übertragen, um den sich ein Roboterarm bewegen soll - z.B. 5° - und die Datensendung kommt drei mal beim Subscriber an, würde sich der Roboterarm um 15° bewegen, was fatale Folgen haben könnte.

Last Will and Testament
Ein Publisher kann festlegen, dass der Broker für den Fall, dass die MQTT-Verbindung verloren geht, statt der abonnierten Werte/Daten eine bestimmte Nachricht an den/die Subscriber sendet.

Retained Message
Mit Setzen dieses Flags weist der Publisher den Broker an, den letzten gesendeten Wert / die letzten Daten zwischenzuspeichern und einem Subscriber, der sich neu verbindet, sofort zu übermitteln.

Alle drei Features sind insbesondere dann sehr nützlich, wenn über nicht immer zuverlässige Übertragungswege (z.B. mobile Netze) übertragen wird.

Eigenschaften und Vorteile von MQTT

MQTT gilt als das Übertragungsprotokoll für das „Internet der Dinge“ und bietet insbesondere für den Datenaustausch über das Internet diverse Vorteile:

- Da alle MQTT nutzenden Endgeräte als Client arbeiten, ist die Überwindung von Firewalls und Security-Maßnahmen meist ohne oder mit überschaubarem Aufwand möglich (es müssen in den Firewalls keine Portfreigaben und kein NAT-Routing / Port-Forwarding eingerichtet werden).
- Der Publisher muss sich nicht darum kümmern, welche Empfänger die bereitgestellten Daten tatsächlich bekommen.
- Anders als bei anderen Internet-basierten Protokollen können Binärdaten ohne Base64- oder andere Kodierung übertragen werden.

Eigenheiten und Nachteile
- Der Datenlieferant bekommt keine Kenntnis darüber, wer seine Daten tatsächlich empfängt (keine Ende-zu-Ende-Quittierung).
- Keine Echtzeitfähigkeit

OPC – Der Prozessdaten-Dolmetscher

Grundsätzliches

In der Automatisierungstechnik werden meist Hardware-Komponenten verschiedenster Hersteller zu einer Anlage zusammengesetzt. Hierbei verfolgte in der Vergangenheit jeder Hersteller einen eigenen Weg, Prozessdaten an die Software-Ebene weiterzugeben. Das betrifft sowohl den physikalischen Kommunikationsweg als auch das Datenformat.

Um diesem Prozessdaten-Babylon zu entgehen, wurde der ursprüngliche OPC-Standard eingeführt. Federführend hierbei war die in 1996 als nicht kommerzielle Organisation gegründete OPC Foundation. Mitglieder der OPC Foundation sind Vertreter führender Unternehmen der Automatisierungsbranche.

Ziel war es, einen weltweit akzeptierten Standard für Kommunikation in der Automatisierungstechnik zu schaffen.

OPC DA - OPC Data Access

OPC DA ist kein Netzwerkprotokoll im eigentlichen Sinne. OPC DA ist aber dennoch ein wichtiger Industriestandard und kann je nach Endgerät auch Berührungspunkte mit Netzwerkkommunikation haben.

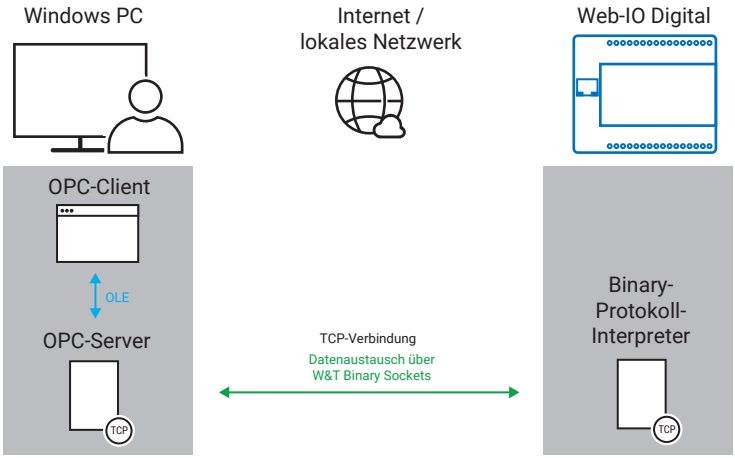
OPC steht für OLE for Process Control, wobei OLE die Abkürzung für Object Linking and Embedding ist. Die Grundidee von OLE ist die geregelte Einbettung von Dokumenten anderer Anwendungen in die eigene Anwendung, zum Beispiel das Einfügen eines Excel-Dokuments in eine Word-Datei.

Sowohl OLE als auch OPC wurden speziell für PCs mit Windows-Betriebssystem konzipiert und funktionieren auch nur auf Microsoft-Betriebssystemen.

OPC-gestützte Anwendungen kommunizieren nicht auf direktem Weg mit den angesprochenen Endgeräten. Stattdessen wird für das entsprechende Endgerät ein OPC-Server installiert. Der OPC-Server ist ein Softwareprozess, der im Hintergrund die herstellerspezifische Kommunikation mit dem Endgerät abwickelt - ähnlich wie es ein Treiber für Hardware auf dem eigenen PC tut. Die so zugänglich gemachten Prozessdaten werden nach dem OPC-Standard aufbereitet und der Anwendung in einer standardisierten Form übergeben.

Der Teil der Anwendung, der mit dem OPC-Server kommuniziert, heißt OPC-Client.

Das folgende Beispiel zeigt den Zugriff auf ein Wiesemann & Theis Web-IO 12xDigital mittels OPC-Server:



Der Zugriff über OPC

Bei der ursprünglichen OPC-Schnittstelle unterscheidet man zwischen vier Hauptaufgaben:

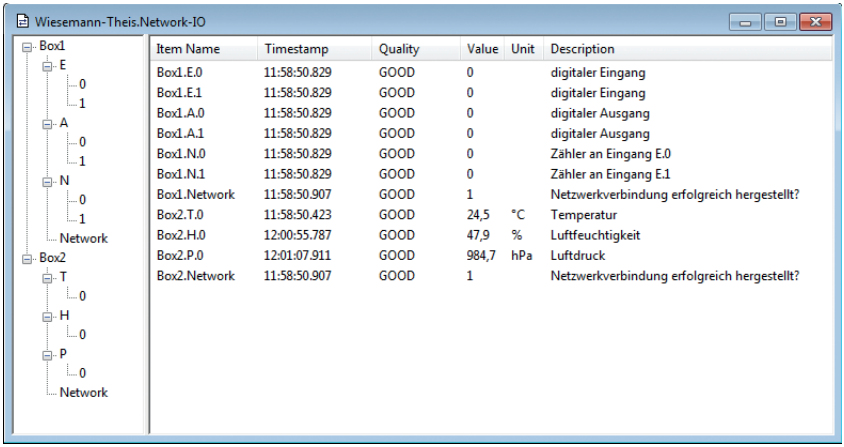
- Data Access: kurz DA, beschreibt den Austausch von Echtzeitdaten über OPC.
- Alarm & Events: kurz AE, dient zur Alarm- und Ereignisbehandlung.
- Historical Data Access: kurz HDA, erlaubt gespeicherte, historische Werte und Werteverläufe zugänglich zu machen.
- Data Exchange: kurz DX, erlaubt es OPC-Servern untereinander Daten auszutauschen.

OPC behandelt die Prozessdaten als einzelne Datenendpunkte. Ein Datenendpunkt kann ein Messwert, ein Prozessstatus, ein Schaltzustand und vieles mehr sein. Die einzelnen Datenendpunkte werden als Items bezeichnet. Die Items können je nach Art gelesen oder geschrieben werden.

Alle Items haben eine Item-ID, eine innerhalb des OPC-Servers einmalige, also eindeutige Adresse. Jedes Item hat eine unbestimmte Anzahl von Properties bzw. Item-Eigenschaften, wie z.B. Wert, Qualität, Zeitstempel, usw.

Die Items werden vom OPC-Server meist in Gruppen zusammengefasst. Daraus ergibt sich dann eine Art Hierarchie (OPC-Server > OPC-Group > OPC Item).

Um dem OPC-Client einen einfachen Zugang zu allen verfügbaren Items zu ermöglichen, erlauben viele OPC-Server dem OPC-Client das OPC-Browsing. Der OPC-Client kann darüber alle Items in einer Art Verzeichnisbaumstruktur abfragen. Hier als Beispiel die Strukturen der Items eines W&T Web-IO 2xDigital und eines Web-Thermo-Hygrobarometers.



Kommunikation zwischen OPC-Client und OPC-Server

Der OPC-Client kann aus allen vom OPC-Server angebotenen Items eine Teilmenge (oder auch alle) auswählen und seinerseits zu einer oder mehreren Gruppen zusammenfassen. Diese Gruppen müssen nicht identisch mit den vom OPC-Server gebildeten Gruppen sein. Die ausgewählten Items werden dann gruppenweise vom OPC-Client abonniert. Das bedeutet, der OPC-Client muss nicht ständig den Zustand der Items abfragen, sondern wird vom OPC-Server automatisch informiert, wenn sich eine der Eigenschaften eines Items ändert. Auf diese Weise entlastet der OPC-Server den OPC-Client und somit die Anwendung.

Wann ist es sinnvoll mit OPC DA zu arbeiten?

Immer dann, wenn eine flexible Anwendung entstehen soll, die ohne großen Aufwand mit der Hardware verschiedenster Hersteller Daten austauschen muss, ist OPC die ideale Lösung.

In Applikationen der Prozessleittechnik und der Prozess- und Messdatenvisualisierung ist OPC vor allem für den Anwender eine feine Sache.

Bei allen Vorteilen der OPC-Technik soll hier aber nicht verschwiegen werden, dass die Programmierung einer universellen OPC-Client-Anwendung eine komplexe

Aufgabe ist, die ein hohes Maß an Programmierkompetenz voraussetzt.

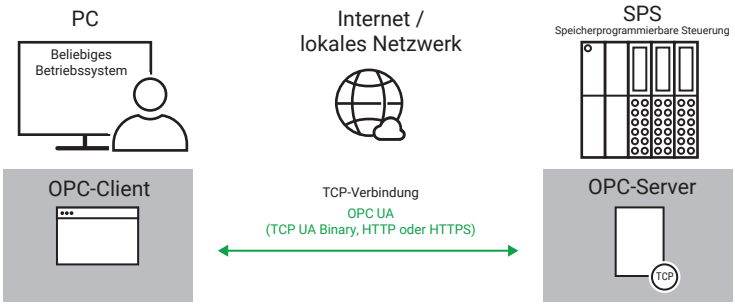
Wenn es also darum geht, eine spezielle Anwendung für ein spezielles Endgerät eines Herstellers zu erstellen, sollte man abwägen, ob es nicht einfacher ist, den direkten, vom Hersteller vorgesehenen, Kommunikationsweg zu gehen.

OPC UA - OPC Unified Architecture

OPC UA ist keine erweiterte Neuauflage des ursprünglichen OPC-Standards. Stattdessen folgt OPC UA einem komplett neuen Konzept und befreit sich dadurch von vielen Nachteilen, die es unter dem ursprünglichen OPC gab.

Das Konzept von OPC UA

Der grundlegendste Unterschied zum ursprünglichen OPC-Standard besteht darin, dass der OPC-Server nicht mehr auf der Client-Seite, sozusagen als zusätzlicher Treiber, installiert werden muss. Stattdessen arbeitet in jedem OPC-UA-fähigen Endgerät der zugehörige OPC-Server.



OPC UA ist:

- plattformunabhängig:
  keine Bindung mehr an Microsoft Betriebssysteme
- skalierbar:
  Systemerweiterungen sind ohne Installation zusätzlicher OPC-Server möglich.
- internetfähig:
  Durch TCP/IP als Basisprotokoll ist OPC UA netzwerkübergreifend einsetzbar.
- sicher:
  OPC UA kann bei Bedarf über eigene Sicherheitsmechanismen oder SSL/TLS abgesichert werden
  Mehr zu SSL/TLS im Kapitel Datensicherheit/Netzwerksicherheit.

Übertragung auf Netzwerkebene

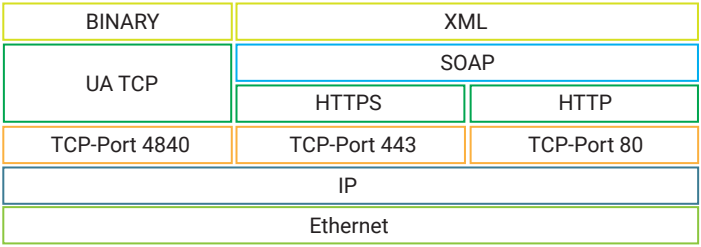
OPC UA arbeitet nach dem Client/Server-Prinzip. Um den OPC-Server ins Endgerät auszulagern, wird auf dem Übertragungsweg zwischen OPC-Client und -Server standardisierte Kommunikation benötigt.

Um das zu gewährleisten, wurde als Basisprotokoll TCP/IP und als physikalischer Standard Ethernet gewählt.

Dabei unterscheidet OPC UA zwischen drei Übertragungsvarianten:

- **HTTP**
  Über HTTP-Requests werden Daten versendet bzw. angefordert.
  Informationen werden SOAP- bzw. XML-formatiert übertragen.
  TCP-Server-Port ist 80.
- **HTTPS**
  Für HTTPS gilt das gleiche wie für HTTP, allerdings arbeitet HTTPS SSL/TLS-verschlüsselt.
  TCP-Server-Port ist 443.
- **UA TCP Binary**
  Die Binary-Variante verzichtet auf den Overhead, der durch die zusätzlichen XML-Tags entsteht. Stattdessen gibt es ein sehr schlankes Protokoll, das den Datenaustausch regelt. Dadurch ist der Datenaustausch deutlich schneller.
  TCP-Server-Port ist 4840.

OPC UA Protokollebenen



Bereits auf Netzwerkebene bietet OPC UA damit sehr flexible Zugriffsmöglichkeiten.

Protokoll- und Anwendungsebene

Neu ist auch, dass OPC UA neben dem Zugriff auf einzelne Items auch komplexe Datenstrukturen zulässt. Außerdem können über OPC UA auf dem Endgerät selbst Programme und Funktionen aufgerufen werden.

Die ursprünglichen Standards OPC DA, EA, HDA und DX sind in OPC UA als mögliche Anwendungsoptionen integriert worden.

Der OPC-UA-Server

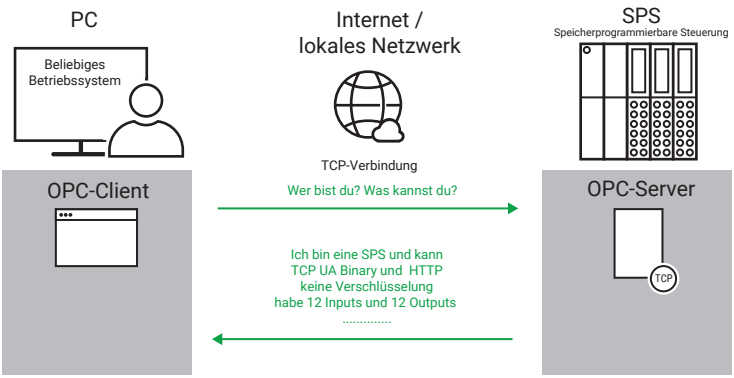
Hauptmerkmal von OPC UA ist wie bereits erklärt, dass der OPC-Server Bestandteil des anzusprechenden Endgerätes ist. Auch wenn OPC UA eine große Vielfalt an Möglichkeiten bietet, muss nicht jeder OPC-Server die gesamte Bandbreite unterstützen. Es reicht, wenn der Server die für die Anwendung benötigte Teilmenge beherrscht.

In Form einer standardisierten Typeninformation fasst der OPC-Server zusammen, welche Möglichkeiten und Protokollvarianten er unterstützt.

Der OPC-UA-Client

Im Gegensatz zum OPC-UA-Server sollte der OPC-UA-Client möglichst viele der verschiedenen Varianten unterstützen. Nur so kann ein hohes Maß an Kompatibilität mit möglichst vielen Endgeräten geschaffen werden.

Der OPC-UA-Client kann aus dem OPC-UA-Server die Typsysteminformation abrufen, in der festgehalten ist, welche Übertragungsverfahren, Items, Variablen, Objekte, Funktionen usw. zur Verfügung stehen. Das vereinfacht die Integration neuer Endgeräte und den damit verbundenen Konfigurationsaufwand erheblich.



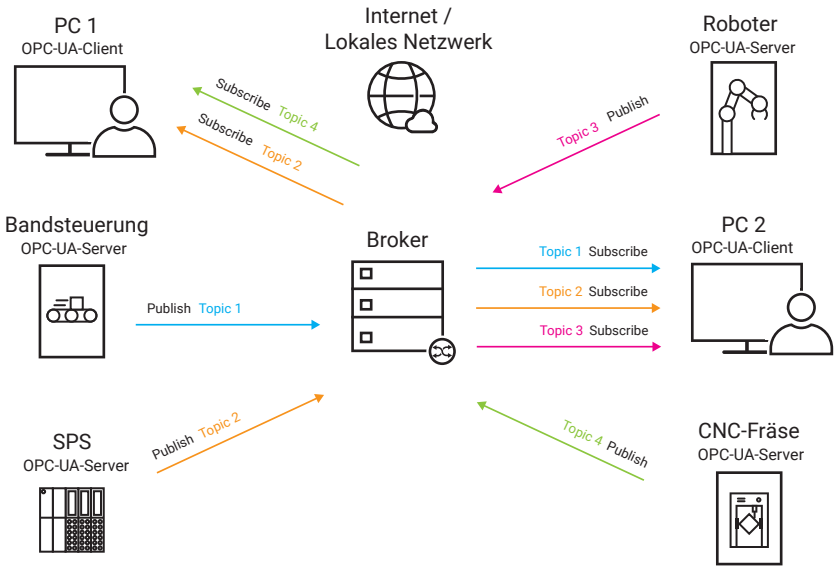
Viele Clients unterstützen neben OPC UA auch noch den ursprünglichen OPC-Standard, so dass auch ein gemischter Betrieb möglich ist. Außerdem gibt es einige Anbieter von OPC-UA-Gateways, die Endgeräte, die von Hause aus kein OPC UA unterstützen, in OPC-UA-Anwendungen integrieren.

OPC UA Pub/Sub

2018 veröffentlichte die OPC-Foundation ein neues Release des OPC-UA-Standards. OPC-UA-Pub/Sub erhält vollständig die Kompatibilität zu OPC UA, unterstützt aber neben der klassischen Client/Server-Kommunikation das Publish/Subscriber - Modell. Dazu nutzt OPC-UA-Pub/Sub die Mechanismen von MQTT.

In Endgeräten, die das Publish/Subscriber-Verfahren von OPC UA nutzen, ist der OPC-Server um einen MQTT-Client-Dienst erweitert. Auch wenn es sprachlich etwas irreführend ist, bleibt die Bezeichnung OPC-Server erhalten.

OPC-Client und OPC-Server können sowohl Daten per Publish an einen Broker senden als auch Daten per Subscribe abonnieren.



So können Prozessdaten mit geringem Aufwand an eine Vielzahl von Endpunkten weitergegeben werden.

Ebenfalls neu bei dem OPC-UA-Pub/Sub Release ist, dass auch UDP als Basisprotokoll erlaubt ist. Da UDP durch einen geringeren Overhead und die verbindungslose Kommunikation über Datagramme schneller ist als TCP, ist die Nutzung von UDP vor allem für Anwendungen vorteilhaft, die auf kurze Reaktionszeiten angewiesen sind.

Mögliche Anwendungsfälle

Bei den ursprünglichen OPC-Anwendungen war es meist so, dass ein OPC-Client als zentrales Leitsystem die beteiligten Endgeräte überwacht und ggf. auch angesteuert hat. Das Leitsystem selbst hatte aber fast immer über Bildschirm und Tastatur eine Zugriffsmöglichkeit für den Anwender.

Neben dieser klassischen Variante unterstützt OPC UA noch folgende Kommunikationsmodelle (ohne Beteiligung eines Leitsystems):

- Endgerät zu Endgerät
- Endgerät zu Datenbank
- Endgerät zu Cloud

Damit ist OPC UA deutlich flexibler als das klassische OPC.

Datensicherheit / Netzwerksicherheit

Grundsätzliches

Sicherheitsanspruch

Wie viel Sicherheit wird bei der Datenübertragung gebraucht? Auf diese Frage gibt es keine pauschale Antwort. Der Anspruch an die Datensicherheit unterscheidet sich von Fall zu Fall und kann je nach Anwendung und Netzwerkumfeld sehr individuell ausfallen.

Aber welche Kriterien müssen erfüllt sein, damit eine Datenübertragung als sicher eingestuft werden kann?

Konkret geht es um drei Anforderungen, die erfüllt sein müssen:

- Integrität
  Sollten die Daten auf dem Transportweg verändert oder manipuliert werden, muss das sofort erkannt werden.
- Vertraulichkeit
  Dritte dürfen die übertragenen Dateninhalte nicht mitlesen können.
- Authentizität
  Es muss sichergestellt sein, dass der Kommunikationspartner tatsächlich der ist, mit dem man die Daten austauschen möchte.

Weitere wichtige Aspekte sind:

- Verfügbarkeit
  Sowohl die von den Kommunikationspartnern angebotenen Dienste als auch die für die Kommunikation benötigte Infrastruktur sollte jederzeit nutzbar sein.
- Beherrschbarkeit
  Auch wenn die für Datensicherheit benötigte Technik sehr komplex ist, sollte sie für Anwender, aber auch für Administratoren beherrschbar bleiben.

In den folgenden Abschnitten werden wir Schritt für Schritt die grundsätzlichen Techniken erklären, die in der Praxis angewendet werden, um diese Punkte zu erfüllen. Abschließend werden wir die Kombination dieser Techniken anhand von HTTPS (sichere Kommunikation im Browser) zusammenfassen.

Begriffe und Symbole

Wir werden uns darauf beschränken, die Prinzipien von Sicherheitsmaßnahmen zu beschreiben, ohne näher die sehr komplizierte Mathematik zu erklären, die dahinter steckt.

Hierbei arbeiten wir mit folgenden Symbolen:

Client

Der Client ist derjenige, der bei der Datenkommunikation die erste Initiative ergreift. Beim Surfen im Internet z.B. der Browser.

Server

Ein Server bietet Datendienste an, die ein oder mehrere Clients nutzen können. So z.B. ein Webserver, von dem Webseiten abgerufen werden können und mit dem der Client Daten austauschen kann.

Kommunikationsdaten

Daten, die zwischen Client und Server ausgetauscht werden

Tresor

Der Tresor symbolisiert durch Verschlüsselung gesicherte Daten.

Symmetrischer Schlüssel zum Ver- und Entschlüsseln

In der Datentechnik steht der Schlüssel für einen Zahlenwert, der zum Ver- und Entschlüsseln genutzt wird. Bei der symmetrischen Verschlüsselung wird für Ver- und Entschlüsselung mit dem gleichen Schlüssel gearbeitet.

Privater Schlüssel
Private Key
zum Verschlüsseln

Öffentlicher Schlüssel
Public Key
zum Entschlüsseln

Bei der asymmetrischen Verschlüsselung wird mit unterschiedlichen Schlüsseln für Ver- und Entschlüsselung gearbeitet. Der Schlüssel zum Verschlüsseln (Verschließen) wird als Vorhängeschloss symbolisiert, das nur mit dem zugehörigen Schlüssel geöffnet werden kann.

Öffentlicher Schlüssel
Public Key
zum Verschlüsseln

Privater Schlüssel
Private Key
zum Entschlüsseln

Außerdem ist immer einer der beiden Schlüssel öffentlich, also für jeden zugänglich - den anderen privaten Schlüssel kennt nur der Besitzer. Ob der Schlüssel zum Verschlüsseln oder der Schlüssel zum Entschlüsseln der öffentliche ist, hängt von der Anwendung ab (mehr dazu im weiteren Verlauf).

Im Folgenden sind zusammengehörende Schlüsselpaare in gleicher Farbe abgebildet.

Hash-Wertberechnung

Eine Art Checksummenberechnung über eine Datenmenge

Hash-Wert

Symbol für den Hash-Wert

Zertifizierungsstelle
Certificate Authority

Zertifizierungsstellen (Certificate Authorities, kurz CAs) stellen Zertifikate aus und bestätigen mit einer Art digitaler Unterschrift die Echtheit.

Zertifikatsdaten

Inhalt des Zertifikates

Digitale Signatur

Die Signatur ist eine Art digitale Unterschrift, mit der die Echtheit eines Zertifikates abgesichert wird.

Signiertes Zertifikat

Signiertes Zertifikat inklusive des öffentlichen Schlüssels des Zertifikatinhabers

Kommunikationsdaten

Was sind eigentlich Daten?

Um zu verstehen, wie Datensicherheit funktioniert, müssen wir uns zunächst noch einmal in Erinnerung bringen, in welcher Form unsere Daten überhaupt codiert und übertragen werden.

Egal ob Text, Webseiten, Bilder, Musik, Videos oder andere Daten übertragen werden sollen - es wird immer eine bestimmte Menge an Bytes von A nach B übertragen.

Bits und Bytes

Zur Erinnerung: Auf unterster Ebene arbeiten Computer mit Bits, also mit Speicherstellen, die den Wert 1 oder 0 haben können. Acht Bits bilden ein Byte.

Ein Byte ist ein Zahlenwert zwischen 0 und 255. In der Datentechnik werden Bytes für gewöhnlich in zweistelliger hexadezimaler Schreibweise dargestellt - also 00 bis FF (siehe Kapitel Zahlensysteme).

Codierung

Je nach Anwendung wird z.B. aus einem Text eine bestimmte Menge an Bytes, wobei jedes Byte einem Buchstaben entspricht.

S	I	C	H	E	R	H	E	I	T	*Text*
53	49	43	48	45	52	48	45	49	54	*Bytes ASCII-kodiert*
Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8	Byte9	Byte10	*(hexadezimal)*

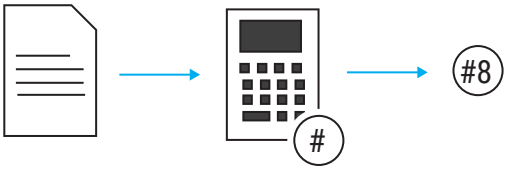
Bei einem Bild wäre in einer Menge an Bytes codiert, welcher Bildpunkt an welcher Position welche Farbe hat.

Welche Bedeutung die einzelnen Bytes in der Anwendung haben, spielt auf dem Transportweg keine Rolle. Hier ist es nur eine entsprechende Menge an Bytes, also Zahlen, mit denen man bei Bedarf Rechenoperationen vornehmen kann.

Integrität - Checksumme und Hash-Wert

Das Prinzip von Hash-Werten

Um sicherzustellen, dass übertragene Daten auf dem Transportweg nicht verändert wurden, versieht der Versender sie mit einer Art Fingerabdruck. In der Praxis erfolgt das, indem die Daten in Ihrer Gesamtheit oder blockweise einer mathematischen Berechnung unterzogen werden. Das Ergebnis wird als Checksumme oder auch Hash-Wert bezeichnet.



Der Trick dabei besteht darin, dass aus der Checksumme die Originaldaten nicht zurück berechnet werden können. Solche Berechnungen werden als Einwegfunktionen bezeichnet.

Ein sehr einfaches Beispiel für Einwegfunktionen ist die Modulo-Berechnung, also die Berechnung des Restwertes bei einer Division.

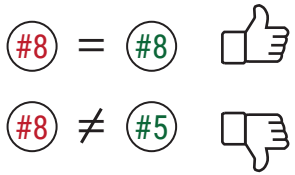
Beispiel:
Den Ursprungswert, z.B. 36, teile ich durch 7. Das ergibt 5 und einen Rest von 1. Die 1 wäre in diesem Fall der Hash-Wert.

Hat man nur den Restwert, ist es unmöglich, eindeutig den Ursprungswert zu benennen, aus dem sich dieser Restwert ergeben hat. Dieses einfache Verfahren ist natürlich ungeeignet, die Integrität von Daten sicherzustellen, da es eine Vielzahl von Ursprungswerten gibt, die einen identischen Hash-Wert ergeben. Deshalb werden bei Sicherheitsprotokollen deutlich kompliziertere Rechenoperationen angewendet.

Hash-Werte in der Praxis

Um sicherzustellen, dass die Daten auf dem Transportweg nicht verändert wurden, berechnet der Versender den Hash-Wert seiner Datensendung und teilt ihn dem Empfänger mit. Der Empfänger berechnet seinerseits den Hash-Wert der empfangenen Daten.

Nur wenn beide Hash-Werte gleich sind, kann davon ausgegangen werden, dass die Daten unverändert angekommen sind.



Im Folgenden werden kurz die gängigsten Standards für Hash-Wertberechnung vorgestellt. Wer sich nicht tiefer mit den Standards auseinandersetzen möchte, kann im nächsten Abschnitt weiterlesen.

- **MD5 - Message Digest Algorithm 5**
  Eine gängige Methode zur Checksummenberechnung ist der MD5-Algorithmus. Der MD5-Algorithmus erzeugt aus einer beliebigen Anzahl von Bytes ein immer 128Bit langes Ergebnis, den MD5-Hash. Wird in der vorliegenden Datenmenge auch nur ein einziges Byte verändert, führt das zu einem komplett anderen MD5-Hash.

  Allerdings ist es mit sehr hohem Rechenaufwand gelungen, verschiedene Ursprungsdaten zu finden bzw. zu generieren, die den selben MD5-Hash ergeben. Nach Bekanntwerden dieser sogenannten Kollisionen gilt MD5 nicht mehr als uneingeschränkt sicher.

- **SHA-1 - Secure Hash Algorithm 1**
  Nachdem Zweifel an der Sicherheit von MD5 aufkamen, entstanden einige konkurrierende Verfahren zur Hash-Werterzeugung, die sich aber allesamt nicht durchgesetzt haben. Mit der immer globaleren Nutzung des Internets wurde dann Anfang der 2000er Jahre SHA-1 als einheitlicher Standard definiert.

  SHA-1 liefert einen 160Bit langen Hash-Wert und gilt bis heute als sicher, auch wenn theoretisch nachgewiesen werden konnte, dass es auch bei SHA-1 zu Kollisionen kommen könnte, wenn genug Rechenleistung zur Verfügung steht.

- **SHA-2, SHA-3 bzw. SHA256**
  Um das Maß an Sicherheit noch einmal zu erhöhen, wurden die Verfahren SHA-2 und später sogar SHA-3 eingeführt. Dabei erlauben beide Standards verschiedene Bit-Tiefen für den Hash-Wert (SHA224, SHA256, SHA384 und SHA512). Gebräuchlich ist SHA256, das mit 256Bit Hash-Werten arbeitet.

Wichtig ist, dass Sender und Empfänger sich auf das gleiche Hash-Verfahren einigen.

Vertraulichkeit durch Verschlüsselung

Vertraulichkeit ist eine der Säulen sicherer Datenübertragung. Dazu müssen die übertragenen Daten auf dem Transportweg vor dem Zugriff Dritter geschützt werden.

Ein wirksamer Schutz vor dem Ausspähen von Daten ist die Verschlüsselung.

Verschlüsselung in der Datentechnik bedeutet eine mathematisch/logische Manipulation der übertragenen Bytes bzw. Zahlenwerte.

Die übertragene Information wird dadurch nur für den gewünschten Empfänger lesbar, der sowohl das Verschlüsselungsverfahren als auch den verwendeten Schlüssel kennt. Alle anderen sehen nur unverständlichen Zahlensalat.

Um das zu erreichen, gibt es zwei Möglichkeiten:

1. **Security by Obscurity**
   Sicherheit durch Unklarheit - Die Daten werden nach einem geheim gehaltenen Verfahren bzw. Algorithmus verschlüsselt. Es können dann nur die Kommunikationspartner Daten austauschen, die das angewandte Verfahren kennen.

   Je nach angewandtem Algorithmus könnte das die gewünschte Sicherheit bieten. Da insbesondere bei Internet-Anwendungen immer häufiger mit quelloffener Software gearbeitet wird, wären auch die angewandten Algorithmen sehr leicht auszuspähen. Für die Anforderungen, die heutige Internet-Anwendungen verlangen, ist *Security by Obscurity* deshalb nicht geeignet.

2. **Full Disclosure**
   Vollständige Offenlegung - Hierbei werden die Daten nach standardisierten Verfahren verschlüsselt. Die Sicherheit der Daten wird dabei erreicht, indem der angewandte Algorithmus neben den Übertragungsdaten geheime Zahlenwerte - sogenannte Schlüssel - miteinbezieht. Die Kommunikationspartner können sich

also auf ein allgemein bekanntes Verfahren einigen und benutzen für Unbeteiligte nicht zugängliche (Zahlen-)Schlüssel.

Mit dieser Methode können auch heutige Internet-Anwendungen mit beliebig vielen Kommunikationspartnern Daten austauschen.

Die gängigen Verschlüsselungsmethoden und der Schlüsselaustausch untereinander werden in den folgenden Abschnitten vorgestellt.

Symmetrische Verschlüsselung

Wie bereits erklärt, funktioniert Verschlüsselung in der Datentechnik durch mathematisch/logische Manipulation der übertragenen Bytes bzw. Zahlenwerte.

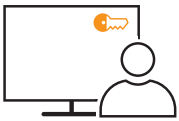
Bei der symmetrischen Verschlüsselung benutzen Sender und Empfänger einen identischen Schlüssel, der sowohl zum Ver-, als auch zum Entschlüsseln genutzt wird.

Das Prinzip symmetrischer Verschlüsselung

Man kann sich das in etwa so vorstellen, als würden wichtige Dokumente beim Transport in einem abschließbaren Tresor verpackt.

Vor dem Versand würde der Tresor vom Versender mit dem passenden Schlüssel abgeschlossen und verschlossen auf den Transportweg gebracht.

Der Empfänger hat den gleichen Schlüssel und kann ihn wieder aufschließen, um an den Inhalt zu gelangen.



Funktionsweise symmetrischer Verschlüsselung

Bei der Übertragung von Daten werden diese natürlich nicht in einen Behälter eingeschlossen.

Eine sehr einfache Verschlüsselung könnte mathematisch so aussehen, dass jedem Byte eine Zahl x aufaddiert wird. Die Zahl x wäre in diesem Beispiel der Schlüssel.

Der Empfänger könnte seinerseits den Schlüsselwert x wieder abziehen, also die Rechenoperation umgekehrt anwenden, um die Originaldaten wiederherzustellen.

Natürlich wäre ein so einfacher Algorithmus viel zu leicht zu knacken und ist deshalb in der Praxis unbrauchbar. Das Beispiel zeigt aber das grundsätzliche Prinzip von Verschlüsselung.

Für echte Verschlüsselung eingesetzte Rechenoperationen sind viel komplexer und manipulieren nicht nur einzelne Bytes.

Als symmetrische Verschlüsselung bezeichnet man Verfahren, bei denen Sender und Empfänger einer Nachricht beim Ver- und Entschlüsseln den gleichen geheimen Schlüssel benutzen, so wie im vorangegangenen Beispiel beschrieben.

Unterschieden wird grundsätzlich zwischen Blockverschlüsselung und Stromverschlüsselung.

Blockverschlüsselung

Bei der Blockverschlüsselung werden die zu übertragenden Daten in gleich große Datenblöcke aufgeteilt. Verbleiben für den letzten Datenblock nicht genug Bytes, werden zusätzliche Füll-Bytes ergänzt.

Jeder Datenblock wird für sich verschlüsselt. Dabei werden die Bytes eines Blocks nicht nur ersetzt (Substitution) sondern auch innerhalb des Blocks in ihrer Position

getauscht (Permutation). Je nach angewendetem Verfahren werden bereits verschlüsselte Datenblöcke mehrmals hintereinander erneut verschlüsselt.

Darüber hinaus unterscheidet die Blockverschlüsselung zwischen drei Verschlüsselungsmodi:

- **ECB-Modus - Electronic Code Book Mode**
  Alle Blöcke werden in gleicher Weise nur mittels des Schlüssel codiert. Die Verschlüsselung von Blöcken gleichen Inhalts führt immer zum gleichen verschlüsselten Code.

- **CBC-Modus - Cipher Block Chaining Mode**
  Neben dem Schlüssel fließen auch Teile vorangegangener Blöcke in die Codierung ein. Die Verschlüsselung von zwei Blöcken gleichen Inhalts führt nicht zum gleichen Ergebnis, wenn die vorangegangenen Blöcke sich unterscheiden.

- **GCM - Galois Counter Mode**
  Neben dem Schlüssel wird ein kontinuierlich hochlaufender Zähler mit in die Verschlüsselung einbezogen. Auch hierbei ergeben zwei Blöcke gleichen Inhalts nach der Verschlüsselung völlig unterschiedliche Daten. Ein Vorteil dieser Technik liegt darin, dass die Blöcke nicht, wie bei CBC, nacheinander verschlüsselt werden müssen. Stattdessen können zeitgleich mehrere Datenblöcke verschlüsselt werden. Hierdurch ist GCM deutlich schneller als CBC.

Bei der symmetrischen Blockverschlüsselung wird fast ausschließlich der GCM-Modus verwendet.

Blockverschlüsselung wird zum Beispiel beim sicheren E-Mail-Versand und der Übertragung von Webseiten mittels HTTPS verwendet. Aber auch auf der Festplatte abgelegte Dateien können blockverschlüsselt sein, um sie vor Fremdzugriff zu schützen.

Stromverschlüsselung

Bei der Stromverschlüsselung wird jedes übertragene Byte einzeln verschlüsselt. Parallel zum Nutzdatenstrom wird ein Schlüsselstrom generiert.

Mit Hilfe des gemeinsamen Schlüssels wird dazu über einen vorgegebenen Algorithmus eine Folge zufällig erscheinender Bytes gebildet. Byte für Byte erfolgt dann eine logisch-mathematische Verknüpfung zwischen Nutzdatenstrom und Schlüssel-

strom. Das daraus resultierende Ergebnis wird übertragen und vom Empfänger umgekehrt wieder entschlüsselt.

Einige Verfahren beziehen in die Berechnung neben dem Schlüsselstrom auch noch bereits übertragene Bytes des Nutzdatenstroms ein.

Stromverschlüsselung wird hauptsächlich bei der Übertragung von analogen Daten und Videosignalen genutzt, da hier das blockweise Verarbeiten zu Stockungen im kontinuierlichen Datenfluss führen könnten.

Symmetrische Verschlüsselungsstandards

Neben einem gemeinsamen Schlüssel müssen sich die beiden Kommunikationspartner zu Beginn des Datenaustauschs einigen, nach welchem Standard verschlüsselt werden soll.

Hier die gängigsten Standards:

- **AES - Advanced Encryption Standard**
  Der Verschlüsselungsalgorithmus AES ist das meistgenutzte Verfahren zur symmetrischen Blockverschlüsselung. Es kann wahlweise mit Schlüssellängen von 128, 192 oder 256 Bit gearbeitet werden. Die möglichen Blockgrößen liegen zwischen 128 und 256 Bit. Jeder Block wird für sich bis zu 14 mal hintereinander verschlüsselt.

  Der verwendete Rijndael-Algorithmus (benannt nach seinem Erfinder) ist ein offener Standard, arbeitet schnell und effizient und ist deshalb auch für weniger leistungsstarke Hardware gut geeignet.

- **DES - Data-Encryption-Standard**
  Bereits in den 70er Jahren wurde DES von IBM entwickelt und wird auch heute noch verwendet, obwohl sich AES inzwischen als Nachfolger durchgesetzt hat.

  DES arbeitet mit 56Bit-Schlüsseln und einer Blockgröße von 64Bit.

  Durch die mit 56Bit eher kurzen Schlüssel gilt DES heute nicht mehr als sicher, da es mit heutigen Computern möglich ist, durch Brute-Force-Attacken (ausprobieren aller Möglichkeiten), den richtigen Schlüssel zu finden.

- **RC4 - Rivest Cipher 4**
  Aufgrund der überschaubaren Codebasis, der einfachen Integration und dem

geringen Ressourcen-Bedarf war RC4 lange eines der beliebtesten Verschlüsselungsverfahren.

Heute gilt die Stromverschlüsselung RC4 als unsicher.

- **ChaCha20**
  ChaCha20 ist eine von Daniel J. Bernstein entwickelte Stromverschlüsselung. Es wird mit einer Schlüssellänge von 256 Bit gearbeitet. Ein Vorteil von ChaCha20 ist die sehr hohe Verschlüsselungsgeschwindigkeit.

- **Twofish**
  Twofish arbeitet mit Blöcken von 128Bit und Schlüssellängen von 128Bit, 192Bit oder 256Bit. Obwohl Twofish als sicher gilt, wird es in der Datenübertragung kaum eingesetzt.

Es gibt noch einige weitere Verschlüsselungsverfahren, die aber heute in der Praxis kaum noch Bedeutung haben.

Preshared Keys
Ein Nachteil der symmetrischen Verschlüsselung ist, dass beide Kommunikationspartner den gleichen gemeinsamen Schlüssel benötigen. Die Schlüssel müssen also vor der Datenübertragung beiden Kommunikationspartnern zugänglich gemacht werden. Das Problem dabei: Wie kann sichergestellt werden, dass beide Seiten den gleichen Schlüssel bekommen, dieser aber gleichzeitig geheim bleibt?

Trotz der Schlüsselproblematik - symmetrische Verschlüsselungsverfahren sind sehr schnell und deshalb besonders für die Übertragung größerer Datenmengen gut geeignet.

Asymmetrische Verschlüsselung

Private und öffentliche Schlüssel
Zur Erinnerung: Bei der symmetrischen Verschlüsselung besteht das größte Sicherheitsproblem darin, dass der gemeinsame Schlüssel geschützt vor Ausspähung zu beiden Kommunikationspartnern gelangen muss.

Asymmetrische Verschlüsselung arbeitet deshalb mit zwei ungleichen Schlüsseln, die aber unzertrennlich zusammen gehören und nur als Schlüsselpaar zusammen einsetzbar sind.

Dabei ist der eine Schlüssel öffentlich, kann von jedem gesehen werden und muss

deshalb auf seinem Weg zum Benutzer nicht weiter vor Ausspähung geschützt werden. Dieser Schlüssel wird als Public Key bezeichnet.

Public Key /
öffentlicher Schlüssel
zum Verschlüsseln

Der andere Schlüssel ist streng geheim und nur seinem Besitzer bekannt.

Niemals darf dieser private Schlüssel aus den Händen gegeben werden.

Private Key /
privater Schlüssel
zum Entschlüsseln

Beide Schlüssel des Schlüsselpaars werden vom Benutzer des privaten Schlüssels generiert. Nur so ist die Geheimhaltung des privaten Schlüssels sichergestellt, da er den Verantwortungsbereich des Besitzers nie verlassen muss.

Weitergegeben wird ausschließlich der Public Key.

Ablauf asymmetrischer Verschlüsselung

Im Fall der asymmetrisch verschlüsselten Datenübertragung wird der Public Key zum Verschlüsseln genutzt. Man kann sich das wie ein Vorhängeschloss vorstellen, das geöffnet verschickt wird und das der Versender zum Verschlüsseln einfach zuschnappt.

Die so verschlüsselten Daten können ausschließlich mit dem zugehörigen Private Key entschlüsselt werden. Der private Schlüssel kann in diesem Fall nicht zum Verschlüsseln genutzt werden.

Es gibt Anwendungsfälle, bei denen es genau umgekehrt ist - dazu im späteren Verlauf mehr.

Im Folgenden wird der prinzipielle Ablauf asymmetrisch verschlüsselter Datenübertragung Schritt für Schritt gezeigt:

Den Public Key - also das geöffnete Vorhängeschloss - fordert der Versender beim Empfänger an.

Der kann den öffentlichen Schlüssel ohne Risiko zum Versender schicken, da er ausschließlich zum Verschließen des Tresors, also zum Sichern der Daten, genutzt werden kann.

Der Versender verschlüsselt seine Daten mit dem Public Key - steckt sie sozusagen in den Tresor - und schnappt das Schloss zu.

Ist das Schloss einmal am Tresor angebracht und zugeschnappt, kommt nicht einmal mehr der Versender an die gesicherten Daten.

Ausschließlich der Empfänger, der im Besitz des streng geheimen Private Key ist, kann die Daten wieder entschlüsseln - also den Tresor mit seinem privaten Schlüssel wieder öffnen.

Damit hat nur der legitime Empfänger Zugriff auf die Originaldaten.

Funktionsweise asymmetrischer Verschlüsselung

Das mathematische Verfahren, das in der realen Datentechnik dahinter steckt, ist deutlich komplizierter.

Auch hier sind die Schlüssel wieder Zahlen, die zwar in einer mathematischen Abhängigkeit zueinander stehen, aber trotzdem kann mit Kenntnis des öffentlichen Schlüssels der private nicht berechnet werden.

Die aufwändigen Algorithmen und die Größe der als Schlüssel gewählten Zahlen führen dazu, dass Ver- und Entschlüsselung sehr rechen- und damit zeitintensiv sind. Eine rein asymmetrische Verschlüsselung ist daher nur für kleinere Datenmengen geeignet. Selbst aktuelle Computerhardware käme bei größeren Datenströmen an die Grenze der verfügbaren Rechenleistung.

Standards zur asymmetrischen Verschlüsselung

Es gibt aktuell drei gängige asymmetrische Verschlüsselungsverfahren:

- **RSA - Rivest, Shamir und Adleman**
  Bereits 1977 entwickelten die Kryptografen Ron Rivest, Adi Shamir und Leonard Adleman das RSA-Verfahren. Die Schlüssellänge ist bei RSA variabel. Für eine sichere Verbindung empfiehlt das Bundesamt für Sicherheit in der Informationstechnik Schlüssel von mindestens 2048Bit.

- **Diffie-Hellman**
  Das Diffie-Hellman Verfahren ist kein Verschlüsselungsverfahren im eigentlichen Sinn. Die Kryptografen Whitfield Diffie und Martin Edward Hellman entwickelten 1976 einen Algorithmus, um zwischen zwei Kommunikationspartnern einen gemeinsamen Schlüssel zu vereinbaren, der dann für eine symmetrische Verschlüsselung genutzt werden kann. Auch wenn man bei diesem Verfahren von Schlüsselaustausch spricht, wird der eigentliche Schlüssel nicht übertragen. Vielmehr gibt der verwendete Algorithmus beiden Seiten die Möglichkeit, den gemeinsamen Schlüssel zu errechnen.

- **ElGamal**
  Das ElGamal-Verschlüsselungsverfahren wurde 1985 vom Kryptologen Taher ElGamal entwickelt und basiert auf den Methoden, die bereits bei Diffie-Hellman genutzt werden. Allerdings wurde der Algorithmus so geändert, dass neben

dem reinen Schlüsselaustausch auch Nutzdaten verschlüsselt und versendet werden können. ElGamal unterliegt keinen Patenten und wird deshalb gerne in Open-Source Projekten genutzt.

Elliptic Curves
Kryptographie auf Basis elliptischer Kurven ist kein eigenes Verschlüsselungsverfahren. Vereinfacht gesehen arbeiten alle asymmetrischen Verfahren mit Einwegfunktionen, die sehr große Primzahlen verarbeiten. Das ist sehr rechenintensiv und damit eben auch sehr zeitintensiv. Laienhaft betrachtet wird anstelle von Primzahlen mit Punkten auf einer elliptischen Kurve gearbeitet. Dieses Verfahren ist deutlich schneller und lässt sich vom Prinzip her auf alle vorgenannten asymmetrischen Verschlüsselungen anwenden.

Hybrid-Verschlüsselung

Um auch große Datenmengen sicher verschlüsselt versenden zu können, werden in der Praxis beide Verfahren kombiniert. Das bedeutet, im ersten Schritt wird das asymmetrische Verschlüsselungsverfahren genutzt, um den geheimen Schlüssel für eine symmetrische Verschlüsselung auszutauschen. Im zweiten Schritt werden die zu übertragenen Nutzdaten mit dem gemeinsamen symmetrischen Schlüssel verschlüsselt.

Hier noch einmal der komplette Ablauf am Beispiel einer vertraulichen Verbindung zwischen Client und Server:

Der Server verfügt über ein Schlüsselpaar zur asymmetrischen Verschlüsselung.

Public Key des Servers /
öffentlicher Schlüssel
zum Verschlüsseln

Private Key des Servers /
privater Schlüssel
zum Entschlüsseln

Der Client erzeugt bei Verbindungsaufbau einen symmetrischen Schlüssel.

gemeinsamer
symmetrischer Schlüssel
zum Ver- und Entschlüsseln

Im ersten Schritt baut der Client eine unverschlüsselte Verbindung zum Server auf und fordert den Public Key des Servers an.

Der Server sendet daraufhin seinen Public Key an den Client.

Der Client verschlüsselt den von ihm generierten symmetrischen Schlüssel mit dem Public Key des Servers.

Geschützt vor Mitlesen durch Dritte wird der Schlüssel sicher zum Server übertragen.

Der Server kann den symmetrischen Schlüssel mit Hilfe seines privaten Schlüssels entschlüsseln.

Nun haben Client und Server den gemeinsamen Schlüssel für die symmetrische Verschlüsselung und können schnell und gesichert Daten austauschen.

Während das asymmetrische Schlüsselpaar des Servers normalerweise immer gleich bleibt, gilt der symmetrische Schlüssel des Clients nur für einen Verbindungszyklus. Für einen späteren Datenaustausch generiert der Client einen neuen Schlüssel.

Schlüsselberechnung nach Diffie-Hellman

Verschlüsselung ohne Schlüsselübertragung

Wie wir bereits festgestellt haben, ist der kritischste Punkt bei der symmetrisch verschlüsselten Datenübertragung, dass die Kommunikationspartner den gleichen geheimen Schlüssel bekommen, ohne dass er auf dem Transportweg ausgespäht wird.

Bei dem Verfahren nach Diffie-Hellman werden weder der gemeinsame Schlüssel, noch Teile davon wirklich übertragen. Stattdessen werden laienhaft gesagt Zahlenwerte ausgetauscht, aus denen jeder Kommunikationspartner auf seiner Seite den gemeinsamen Schlüssel ausrechnen kann.

Funktionsweise von Diffie-Hellman

Für das Diffie-Hellman-Verfahren werden eine möglichst große Primzahl P und eine positive Ganzzahl G benötigt, die kleiner sein muss als P.

- (P) sehr große Primzahl
- (G) kleinere Ganzzahl

Dieses Zahlenpaar benötigen beide Kommunikationspartner. Im Allgemeinen wird es vom Server an den Client übermittelt, der es bei Verbindungsbeginn anfordert.

P und G sind nicht geheim und können bedenkenlos auch über ungesicherte Kanäle weitergegeben werden.

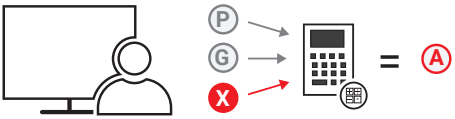
Zusätzlich generiert jeder der beiden Kommunikationspartner eine weitere Ganzzahl, die nur er selbst kennt (X und Y).

Y geheime Ganzzahl vom Server generiert

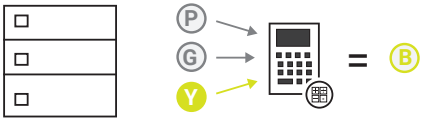
X geheime Ganzzahl vom Client generiert

Damit haben beide Kommunikationspartner jeweils drei Zahlen: zwei öffentliche und eine geheime.

Nun berechnet der Client nach einem vorgegebenen Einwegalgorithmus aus den öffentlichen Zahlen P und G und seiner geheimen Zahl X eine weitere Zahl A.

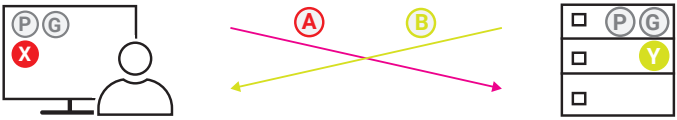


Der Server benutzt den selben Algorithmus um eine Zahl B zu berechnen.



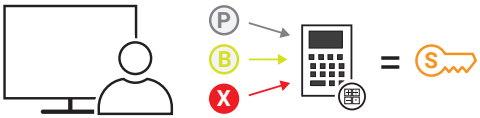
Da es sich um einen Einwegalgorithmus handelt, kann aus den öffentlichen Zahlen P und G und dem Ergebnis A bzw. B die geheime Zahl X bzw. Y nicht eindeutig zurückgerechnet werden.

Client und Server können also A und B ungeschützt miteinander austauschen.

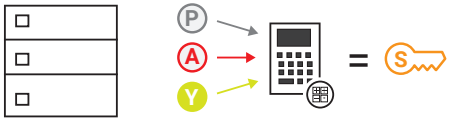


Damit haben Client und Server jeweils vier Zahlen von denen drei zur weiteren Verarbeitung benötigt werden.

Der Client errechnet nach einem weiteren vorgegebenen Algorithmus aus den öffentlichen Zahlen P und B, sowie seiner geheimen Zahl X den gemeinsamen Schlüssel S.



Der Server verfährt mit P und A, sowie seiner geheimen Zahl Y nach dem gleichen Algorithmus.



Die genutzten Algorithmen sind so aufgebaut, dass beide Seiten am Ende zum gleichen Ergebnis und damit zum gleichen gemeinsamen, geheimen Schlüssel kommen.

Der so berechnete Schlüssel wird im weiteren Verlauf der Verbindung für die symmetrisch verschlüsselte Datenübertragung verwendet. Alle bis dahin genutzten Zahlen werden für die eigentliche Verschlüsselung nicht mehr benötigt.

Die Mathematik hinter Diffie-Hellman

In diesem Abschnitt wollen wir anhand eines einfachen Beispiels die Algorithmen zeigen, die bei der Diffie-Hellman-Schlüsselberechnung genutzt werden. Wir verzichten bewusst auf eine detaillierte Erklärung der gezeigten Formeln.

Beim Diffie-Hellman-Schlüsselaustausch wird mit extrem großen Primzahlen gearbeitet. Nur so ist sichergestellt, dass der berechnete Schlüssel nicht durch Ausprobieren mit verschiedenen Zahlenwerten herausgefunden werden kann.

In unserem Beispiel arbeiten wir mit den kleinsten möglichen Werten, was natürlich in der Praxis nicht sicher wäre.

P = 5	Primzahl
G = 4	kleinere Ganzzahl
X = 3	geheime Ganzzahl des Clients
Y = 2	geheime Ganzzahl des Servers

Zunächst berechnen Client und Server aus den öffentlichen Zahlen P und G sowie der geheimen Zahl X bzw. Y die Zahlen A bzw. B.

Die Formel für den Client:

$$A = G^{X} \bmod P$$

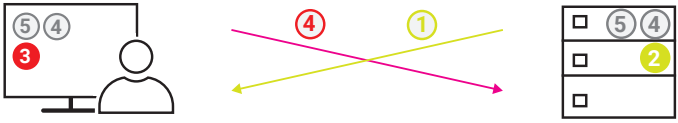
$$A = 4^{3} \bmod 5 = 4$$

Zur Erinnerung: mod steht für Modulo und ist die Restberechnung bei der Division

Die Formel für den Server:

$$B = G^{Y} \bmod P$$
$$B = 4^{2} \bmod 5 = 1$$

Die beiden errechneten Zahlen werden ausgetauscht.



Nun berechnen beide Seiten unabhängig voneinander den gemeinsamen und geheimen Schlüssel.

Die Formel für den Client:

$$S = B^{X} \bmod P$$
$$S = 1^{3} \bmod 5 = 1$$

Die Formel für den Server:

$$S = A^{Y} \bmod P$$
$$S = 4^{2} \bmod 5 = 1$$

Beide Berechnungen führen zum selben Ergebnis. In diesem Beispiel wäre der gemeinsame Schlüssel für die nun beginnende symmetrische Verschlüsselung 1.

Diffie-Hellman zusammengefasst

Das Diffie-Hellman-Verfahren ist kein Verschlüsselungsverfahren im eigentlichen Sinn. Die Kryptografen Whitfield Diffie und Martin Edward Hellman entwickelten 1976 einen Algorithmus, um zwischen zwei Kommunikationspartnern einen gemeinsamen Schlüssel zu vereinbaren, der dann für eine symmetrische Verschlüsselung

genutzt werden kann. Auch wenn man bei diesem Verfahren von Schlüsselaustausch spricht, wird der eigentliche Schlüssel nicht übertragen. Vielmehr gibt der verwendete Algorithmus beiden Seiten die Möglichkeit, den gemeinsamen Schlüssel zu errechnen.

Diffie-Hellman Elliptic Curves

Diffie-Hellman-Schlüsselberechnung auf Basis elliptischer Kurven ist kein eigenes Verfahren. Laienhaft betrachtet wird anstelle von Primzahlen mit Punkten auf einer elliptischen Kurve gearbeitet. Dieses Verfahren ist nicht so rechenintensiv und dadurch etwas schneller als die klassische Diffie-Hellman-Schlüsselberechnung.

Authentisierung durch Zertifikate

Zunächst die Definition von drei Begriffen, die oft verwechselt werden:

Authentisierung
bedeutet einen Nachweis über die eigene Identität zu erbringen.

Authentifizierung
heißt die Identität eines anderen auf Richtigkeit zu prüfen.

Autorisierung
ist die Übertragung von Rechten, die innerhalb eines Systems nur bestimmten Personen oder Institutionen vorbehalten sind.

In den vorangegangenen Abschnitten haben wir Möglichkeiten kennengelernt, Daten beim Transport vor Veränderung und Mitlesen zu schützen. Aber was nutzt diese Sicherheit, wenn der Kommunikationspartner, mit dem ich Daten austausche, gar nicht der ist, mit dem ich eigentlich kommunizieren möchte? Wichtige Daten landen ggf. beim falschen Empfänger.

Dieses fehlende Stück Sicherheit kann nur erreicht werden, wenn der Kommunikationspartner ganz klar zu identifizieren ist.

Wenn ich im richtigen Leben wissen möchte, mit wem ich es zu tun habe, frage ich mein Gegenüber nach seinem Personalausweis. Der ist von einer vertrauenswürdigen Behörde ausgestellt und wenn Bild, Name und Adresse die richtige Person ausweisen, ist alles in Ordnung.

Zertifikate

In der Datenübertragung gibt es auch solche Ausweise, die allerdings als Zertifikat bezeichnet werden.

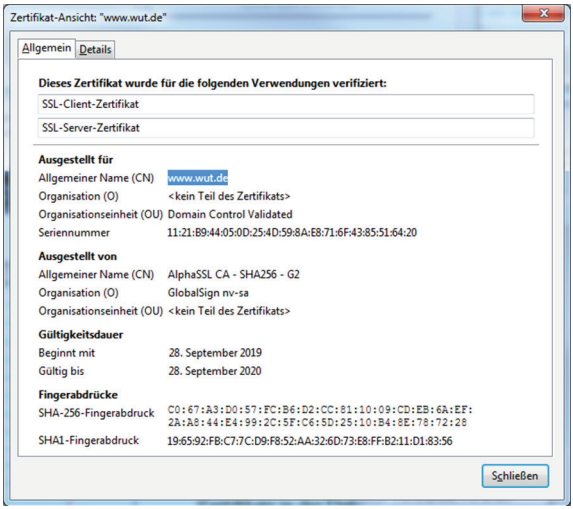
Zertifikatnehmer

Wer benötigt ein Zertifikat?

Alle Institutionen und Personen, die sichere Datendienste wie z.B. Webserver betreiben möchten, benötigen ein entsprechendes Zertifikat, um sich beim Nutzer identifizieren zu können.
Welche Angaben beinhalten Zertifikate?

- Zertifikatnehmer (für wen ist das Zertifikat ausgestellt?)
- Name oder IP-Adresse des Servers, für den das Zertifikat gilt
- Öffentlicher Schlüssel des Zertifikatnehmers
  (um später zu einer sicher verschlüsselten Kommunikation zu wechseln)
- Verwendungszweck
  (für welchen Datendienst wurde das Zertifikat ausgestellt?)
  - Webserver-Authentisierung
  - Software-Signatur
  - Ausstellungsberechtigung für Zwischenzertifikate
  - Authentisierung von Mailservern
- Je nach Anwendung
  - IP-Adresse
  - Domain- bzw. Hostname
- Seriennummer
- Verfallsdatum (wie lange ist es gültig?)
- Zertifikatgeber (wer hat es ausgestellt?)
- Signatur des Zertifikatgebers

Hier als Beispiel die Zertifikatsinformationen des Wiesemann & Theis Webservers:



Zertifikatgeber

Zertifikate in der Datenübertragung werden natürlich nicht vom Einwohnermeldeamt ausgestellt wie ein Personalausweis.

Stattdessen gibt es vertrauenswürdige Zertifizierungsstellen - Certificate Authorities oder kurz CAs.

Zertifizierungsstelle
Certificate Authority

Das Ausstellen, der Aufbau von Zertifikaten und wie sie weiter genutzt werden, ist im ITU-T Standard X.509 festgehalten. Man spricht deshalb auch oft von X.509-Zertifikaten.

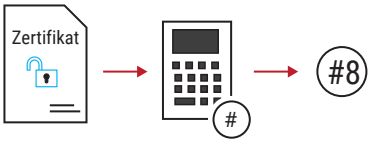
X.509-Zertifikate sind nur dann gültig, wenn sie vom Aussteller signiert - also mit einer digitalen Unterschrift (Signatur) bzw. einem digitalen Fingerabdruck versehen sind.

Signieren von Zertifikaten

Die CAs signieren Zertifikate in zwei Schritten:

Schritt 1

Über den gesamten Zertifikatsinhalt wird nach einem ausgewählten mathematischen Verfahren ein Hash-Wert gebildet.



Schritt 2

Die CA verschlüsselt den Hash-Wert und hängt ihn an das Zertifikat an.

Die Verschlüsselung des Hash-Werts erfolgt asymmetrisch. Allerdings ist es in diesem Fall so, dass mit dem privaten Schlüssel des CA verschlüsselt wird und der öffentliche Schlüssel zur Entschlüsselung benutzt wird.

Privater Schlüssel der CA
Private Key

Zur Erinnerung: Bei der normalen asymmetrischen Verschlüsselung war es genau anders herum - öffentlicher Schlüssel verschlüsselt / privater Schlüssel entschlüsselt.

Der Zertifikatnehmer bekommt das signierte Zertifikat, das aus den unveränderten Zertifikatsdaten und der Signatur, also dem verschlüsselten Hash-Wert besteht.

Verteilung und Gültigkeit von Zertifikaten

Eine entscheidende Voraussetzung für die durch X.509-Zertifikate geschaffene Sicherheit ist, dass der private Schlüssel (private Key) einer CA geheim bleibt und niemals nach außen dringt. Deshalb sind die CAs auch nicht mit dem Internet verbunden, um die privaten Schlüssel vor Ausspähung zu schützen.

Im Gegensatz dazu müssen die öffentlichen Schlüssel der Root-CAs (public Keys) möglichst einfach für alle Nutzer zugänglich gemacht werden.

Das geschieht dadurch, dass die Herausgeber von Software und Betriebssystemen häufig benötigte Schlüssel bereits mit einprogrammiert bzw. in einem Trust Store (Sicheren Speicher) hinterlegt haben.

Darüber hinaus bieten Anwendungen und Betriebssysteme meist die Möglichkeit, Zertifikate nachzuladen und diese als vertrauenswürdig anzuerkennen.

Das Einspielen und Anerkennen von Zertifikaten sollte mit größter Sorgfalt erfolgen.

Die Identität eines Servers oder einer Anwendung ist aber nicht allein durch die korrekte Signierung des zugehörigen Zertifikats sichergestellt. Im nächsten Schritt müssen die weiteren Daten des Zertifikates geprüft werden.

Der Ablauf bei der Nutzung von X.509-Zertifikaten sieht grob so aus:

- Der Zertifikatnehmer beantragt ein Zertifikat und stellt dem Zertifikatgeber, also der CA, alle benötigten Informationen zur Verfügung.
- Die CA prüft, ob alles seine Richtigkeit hat.
- Ist alles in Ordnung, ergänzt die CA die Daten des Zertifikatnehmers um weitere Daten wie die eigene Identität, das eingesetzte Hash-Verfahren, eine eindeutige Seriennummer usw.
- Abschließend signiert die CA das Zertifikat mit ihrem eigenen privaten Schlüssel.

Letztlich geht es ja darum, die öffentlichen Schlüssel von Dienstanbietern bzw. Servern mit einem sicheren Herkunftsnachweis an den Client zu übergeben, der solche Dienste nutzen möchte. Es geht also um die Verteilung der Public Keys.

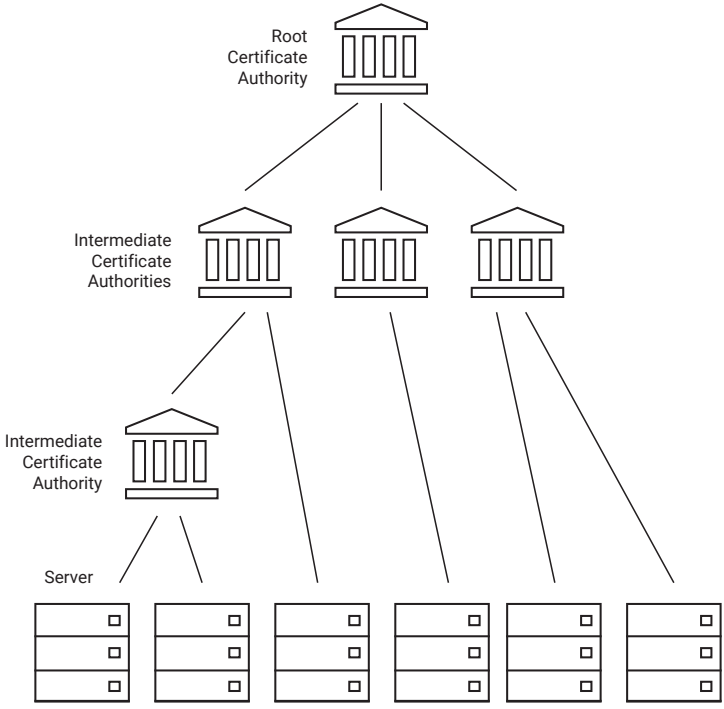
Hierarchie der Zertifizierungsstellen

Durch die rasante Ausbreitung des Internets und der darin angebotenen Dienste ist der Bedarf an signierten Zertifikaten riesig. Eine einzige vertrauenswürdige Zertifizierungsstelle könnte dieser Flut gar nicht Herr werden. Gäbe es aber beliebig viele

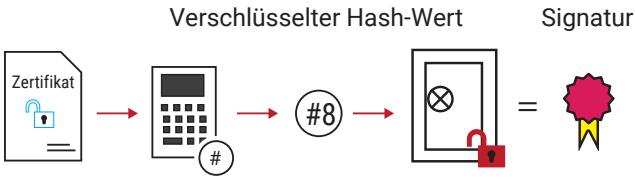
Zertifizierungsstellen, ginge schnell die Kontrolle und damit die Vertrauenswürdigkeit verloren.

Deshalb gibt es letztlich eine begrenzte Zahl vertrauenswürdiger Stamm- oder auch Wurzelzertifizierungsstellen. Diese Root Certificate Authorities schenken untergeordneten Zwischenzertifizierungsstellen ihr Vertrauen.

Diese von den CAs autorisierten Intermediate CAs können ihrerseits Zertifikate für Server (und Clients) erstellen. Sie können aber auch weitere Intermediate CAs berechtigen Zertifikate auszustellen. So ergibt sich eine baumartige CA-Hierarchie.

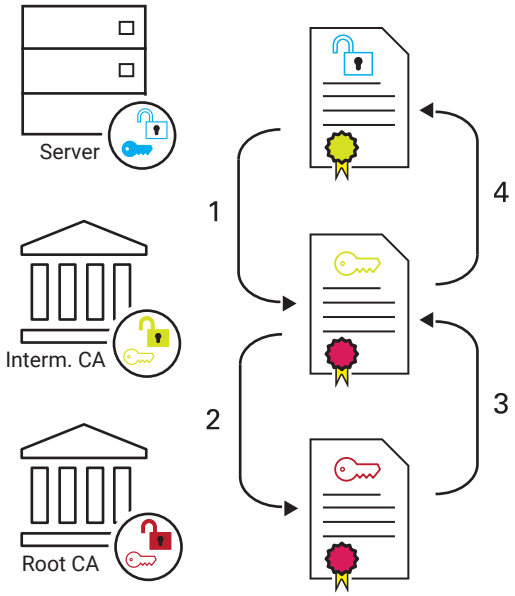


Noch mal zur Erinnerung: Eine Signatur ist der mit dem privaten Schlüssel des Ausstellers verschlüsselte Hash-Wert des Zertifikatinhalts.



Jede CA signiert die ausgestellten Zertifikate für die untergeordnete Intermediate CA mit ihrem privaten Schlüssel. Daraus ergeben sich Zertifikatsketten - Certificate Chains - die zurückführen bis auf die Ebene der Root CA. Da es keine Ebene oberhalb der Root CA gibt, stellt diese sich ein eigenes selbstsigniertes Zertifikat aus.

Der Zertifikatsnehmer (z.B. ein Server) am Ende dieser Kette muss immer alle beteiligten Zertifikate bis hin zum Wurzelzertifikat bereitstellen.



Wenn der Client das Zertifikat des angesprochenen Servers noch nicht kennt, kann er so die Zertifikatskette bis zum Wurzelzertifikat der Root CA zurückverfolgen (1 und 2).

Ist die Root CA als vertrauenswürdig bekannt, kann beginnend mit dem öffentlichen Schlüssel der Root CA die Kette bis hin zum Server-Zertifikat auf Vertrauenswürdigkeit geprüft werden (3 und 4).

Erst nach erfolgreicher Überprüfung kann der innerhalb des Zertifikates übermittelte öffentliche Schlüssel des Server für die anstehende Kommunikation sicher benutzt werden.

Die Public Key Infrastructure
Um Zertifikate zu erstellen, zu verwalten, zu verteilen und ggf. auch wieder zu widerrufen, wird eine Infrastruktur benötigt, die das gemäß der bestehenden Richtlinien und Standards sicher ermöglicht.

Eine Public Key Infrastructure, kurz PKI, besteht aus mindestens einer Wurzelzertifizierungsstelle (Root CA) und je nach Bedarf diversen Unterzertifizierungsstellen. Ferner wird eine Registrierungsinstanz (Registration Authority, kurz RA) benötigt. Alle innerhalb der PKI signierten und ausgegebenen Zertifikate werden von der RA gelistet und verwaltet.

Zertifikate werden von der RA auf Gültigkeit geprüft und können, wenn nötig, auch widerrufen werden.

Würde z.B. der private Schlüssel einer CA in fremde Hände geraten - was eigentlich nicht passieren sollte - würde die zuständige RA alle von dieser CA ausgestellten Zertifikate widerrufen.

Betriebssysteme oder Anwendungen wie Browser oder Mailclient müssen deshalb regelmäßig, z.B. über Updates, die hinterlegten Zertifikate über die PKI mit der RA abgleichen.

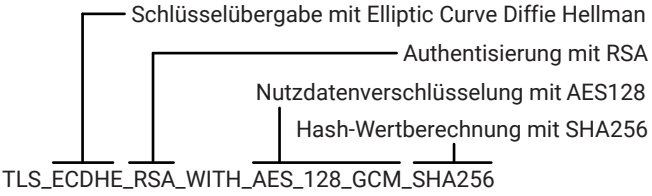
Im Internet gibt es einige anerkannte, vertrauenswürdige PKIs, deren CAs/RAs Zertifikate ausstellen und verwalten. In aller Regel erfolgt das gegen Gebühr. Für große Konzerne, Organisationen oder Behörden kann es sich deshalb lohnen, eine eigene Inhouse Public Key Infrastructure zu betreiben.

Cipher Suites

Wie wir bis hierher erfahren haben, wird in der Datenübertragung mit symmetrischer und asymmetrischer Verschlüsselung gearbeitet. Für beide Verschlüsselungstechniken gibt es unterschiedliche Algorithmen. Das gilt ebenfalls für die Hash-Wertberechnung.

Bevor eine verschlüsselte Datenübertragung beginnt, müssen sich die Kommunikationspartner auf eine Kombination der verwendeten Verfahren einigen.

Diese Kombinationen bezeichnet man auch als Cipher Suites und sie enthalten folgende Informationen:



Die Cipher Suites werden nicht im hier gezeigten Klartext übertragen, sondern jede nutzbare Kombination hat eine eindeutige 2-Byte-Identifikationsnummer. Die hier gezeigte zum Beispiel 0xC02F.

Im Zuge des Verbindungsaufbaus sendet der Client eine Liste mit ihm möglichen Cipher Suites an der Server.

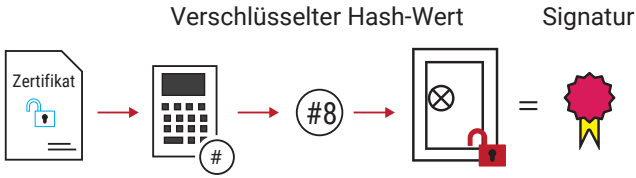
```
Session ID Length: 32
Session ID: bc982a3eae526031c7e612862ede715394547ac10e46b46a...
Cipher Suites Length: 28
Cipher Suites (14 suites)
   Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
   Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
   Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
   Cipher 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
```

Der Server antwortet dem Client, welche Kombination genutzt werden soll.

Prüfen von Zertifikaten

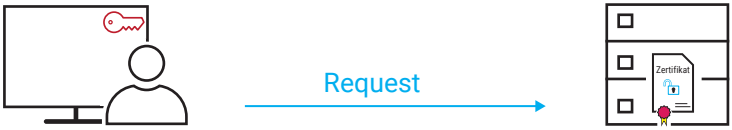
Bis hierhin haben wir sehr grob erklärt, wie die Authentisierung über Zertifikate prinzipiell funktioniert. Da die Zertifikate einen sehr wesentlichen Teil der Sicherheit im Internet ausmachen, lohnt es, noch einmal genauer zu beleuchten, wie die Überprüfung der Zertifikate erfolgt.

Noch mal zur Erinnerung: Zum Signieren verschlüsselt der Aussteller, also die CA, den Hash-Wert des Zertifikatinhalts mit dem privaten Schlüssel.



Der Zertifikatnehmer bekommt das signierte Zertifikat, das aus den unveränderten Zertifikatsdaten und der Signatur, also dem verschlüsselten Hash-Wert, besteht.

Der Kommunikationspartner (in diesem Fall der Client), der eine sichere Authentifikation seines Gegenübers wünscht, fordert das signierte Zertifikat an.



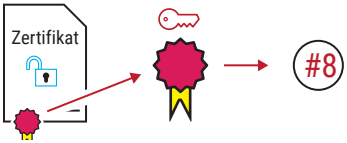
Das Zertifikat wird vom Server umgehend zurückgesendet.

So bekommt der Client die Zertifikatsdaten und den zu den Daten passenden, verschlüsselten Hash-Wert.

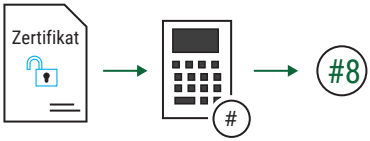
Zum privaten Schlüssel der CA gibt es auch einen passenden öffentlichen Schlüssel. Bei verbreiteten Standardanwendungen wie z.B. Webbrowsern werden die öffentlichen Schlüssel der bekannten CAs in der Software integriert mitgeliefert (auf die Verwaltung der öffentlichen Schlüssel gehen wir im weiteren Verlauf noch näher ein).

Öffentlicher Schlüssel der CA
Public Key

Mit Hilfe des öffentlichen Schlüssels der CA entschlüsselt der Client den von der CA verschlüsselten Hash-Wert wieder.



Parallel dazu berechnet der Client, genau wie es die CA getan hat, den Hash-Wert der Zertifikatsdaten.



Nun vergleicht der Client den entschlüsselten Hash-Wert mit dem selbst errechneten. Sind beide Werte gleich, ist die Echtheit des Zertifikates bestätigt

Unterscheiden sich die beiden Werte, handelt es sich nicht um das Original-Zertifikat und der Client sollte den Kontakt zum Server sofort abbrechen.

Bevor der Client mit dem eigentlichen Datenaustausch beginnt, prüft er außerdem anhand der Einträge, ob das Zertifikat nicht z.B. abgelaufen ist.

SSL/TLS

Geht es um verschlüsselte Datenkommunikation, tauchen sehr schnell die Begriffe SSL und TLS auf.

SSL (Secure Socket Layer) war eine Entwicklung der Firma Netscape und wurde 1995 das erste Mal in der Version 2.0 der Öffentlichkeit vorgestellt. Bereits ein Jahr später folgte Version 3.0.

TLS (Transport Layer Security) ist eine Weiterentwicklung von SSL und hat dieses 1999 abgelöst.

Die Arbeitsweise von SSL und TLS ist in vielen Teilen identisch, weshalb auch meist von SSL/TLS gesprochen wird.

SSL fasst die bis hierher beschriebenen Sicherheitsmechanismen zu einem Protokollablauf zusammen. Hier noch einmal kurz zusammengefasst das Ineinandergreifen der verschiedenen Techniken:

1. Der Client - z.B. ein Browser - baut eine TCP-Verbindung zum Server auf.
2. Wenn die Verbindung zustande gekommen ist, teilt der Client dem Server mit, welche Verschlüsselungsverfahren er unterstützt bzw. benutzen möchte. Das erfolgt noch unverschlüsselt.
3. Der Server antwortet - ebenfalls noch unverschlüsselt - und gibt bekannt, für welches Verschlüsselungsverfahren er sich entscheidet.
4. Darüber hinaus übermittelt der Server sein Zertifikat.
5. Der Client überprüft anhand des Zertifikats die Identität des Servers und stellt damit sicher, dass er mit dem richtigen Kommunikationspartner verbunden ist.
6. Wenn die Identität des Servers bestätigt ist, entnimmt der Client dem Zertifikat den öffentlichen Schlüssel (Public Key) des Servers.
7. Der Client generiert einen gemeinsamen Schlüssel für eine symmetrische Verbindung und verschlüsselt diesen mit dem öffentlichen Schlüssel des Servers.

8. Der Server entschlüsselt die Datensendung des Clients mit seinem privaten Schlüssel und bekommt so den gemeinsamen Schlüssel für den symmetrisch verschlüsselten Datenaustausch.
9. Der so verschlüsselte Datenaustausch erfolgt solange, bis einer der beiden Kommunikationspartner die Verbindung beendet.

HTTPS - SSL/TLS in der Praxis

Am Beispiel des Aufrufes einer Webseite im Browser über HTTPS soll hier der Ablauf einer verschlüsselten Verbindung noch einmal verdeutlicht werden.

Als Adresse wird im Browser eingegeben:

Der Browser baut eine Verbindung zum WuT-Webserver auf und sendet einen HTTPS-Request. In diesem Request teilt er dem Server mit, welche Cipher Suites unterstützt werden.

```
Session ID Length: 32
Session ID: bc982a3eae526031c7e612862ede715394547ac18e46b48a...
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
```

Der Server wählt eine der angebotenen Cipher Suites aus und teilt sie dem Browser mit.

```
Version: TLS 1.2 (0x0303)
Random: 72e4230fcce518518314701fdbcc007f4c95601fa63b0cei...
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)
```

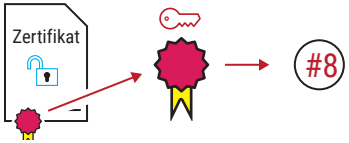
Anschließend übermittelt der Server sein Zertifikat.

So bekommt der Browser die Zertifikatsdaten und kann zunächst prüfen, ob das Zertifikat nicht abgelaufen ist oder widerrufen wurde.

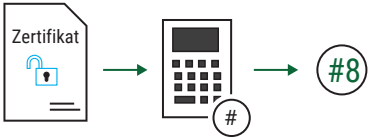
Mit Hilfe des öffentlichen Schlüssels der CA

Öffentlicher Schlüssel der CA
Public Key

entschlüsselt der Browser den von der CA verschlüsselten und dem Zertifikat angehängten Hash-Wert.



Parallel dazu berechnet der Browser den Hash-Wert der Zertifikatdaten, genau wie es die CA getan hat.

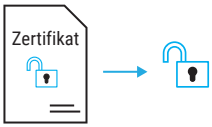


Nun vergleicht der Browser den entschlüsselten Hash-Wert mit dem selbst errech-

neten. Sind der entschlüsselte und der selbst berechnete Hash-Wert gleich, ist die Echtheit des Zertifikates bestätigt.

Damit ist sichergestellt, dass der Browser mit dem richtigen Server verbunden ist.

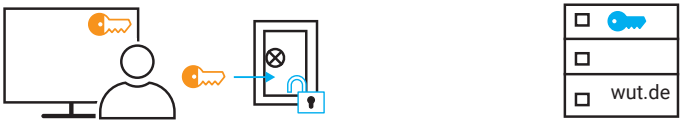
Jetzt kann der Browser dem Zertifikat den Public Key des Servers entnehmen.



Der Browser generiert einen gemeinsamen Schlüssel, der im späteren Verlauf der Browser-Session genutzt wird, um Daten symmetrisch verschlüsselt auszutauschen.

gemeinsamer
symmetrischer Schlüssel
zum Ver- und Entschlüsseln

Diesen gemeinsamen Schlüssel verschlüsselt der Browser mit dem öffentlichen Schlüssel des wut.de-Servers



und schickt ihn zum wut.de-Server.

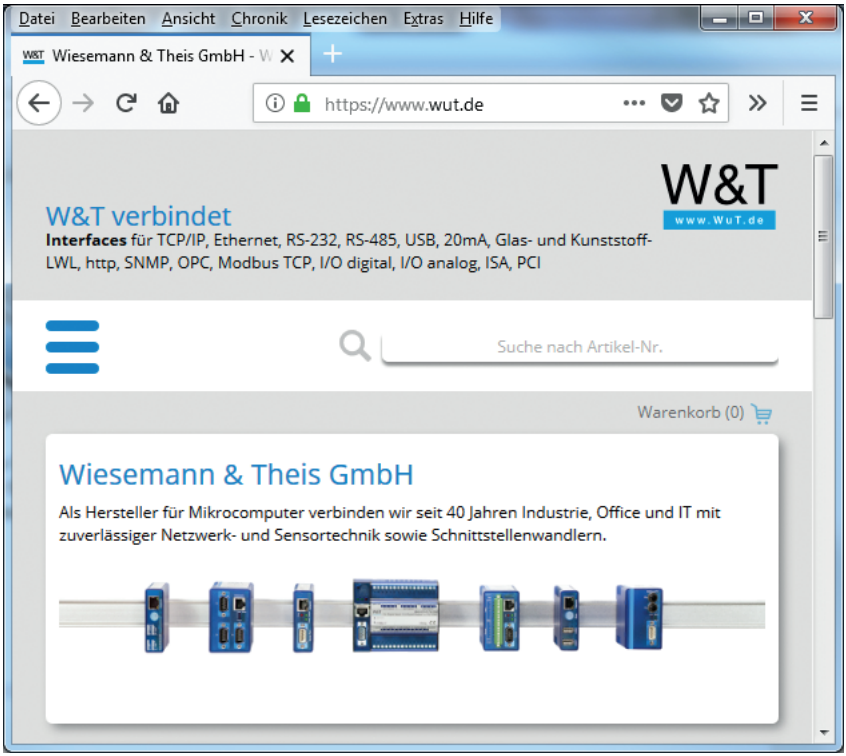
Der wut.de-Server kann die Datensendung mit dem privaten Schlüssel entschlüsseln

Private Key des Servers /
privater Schlüssel
zum Entschlüsseln

und bekommt so den gemeinsamen Schlüssel, ohne dass andere diesen Schlüssel auf dem Übertragungsweg hätten ausspähen können.

An dieser Stelle beginnt der eigentliche Austausch von Nutzdaten. Sowohl der Browser als auch der Server können die übertragenen Daten mit dem gleichen Schlüssel ver-, und entschlüsseln.

So werden alle zum Anzeigen der Webseite benötigten Daten verschlüsselt übertragen.



Das Vorhängeschloss in der Adresszeile zeigt dem Anwender, dass er mit dem richtigen Server verbunden ist.

VPN - Virtual Private Network

Grundsätzliches

Vorweg sei gesagt: Der Einsatz und die Realisierung von VPN erlauben diverse Varianten. Alle Details von VPN zu beleuchten, bietet genug Stoff für ein eigenes Buch und würde den Rahmen dieses Kapitels sprengen. An dieser Stelle sollen deshalb nur die globale Funktion und die wichtigsten Grundbegriffe von VPN vorgestellt werden.

VPN beschreibt die Technik, vertrauliche Netzwerkteile an verschiedenen Standorten über das Internet, also ein öffentliches Netz, miteinander zu verbinden. Typische Beispiele für den Einsatz von VPN sind:

- **Standortübergreifende Netzwerkverbindungen**
  Zwei oder mehr Firmenstandorte können zu einer gemeinsam nutzbaren Netzwerkinfrastruktur zusammengefasst werden.

- **Mitarbeiter im Außendienst**
  Für Mitarbeiter, die außerhalb des Firmenstandorts arbeiten, kann ein Zugang zum firmeninternen Netzwerk oder zu Teilen davon eingerichtet werden. Der Zugriff ist dann z.B. aus dem Netzwerk des Kunden, über öffentliche Hotspots oder das Mobilfunknetz möglich.

- **Heimarbeitsplatz**
  Wie auch für den Mitarbeiter im Außendienst kann einem Mitarbeiter, der zu Hause arbeitet, ein VPN-Zugang eingerichtet werden, der den Zugriff auf das Firmennetzwerk aus dem Home-Office erlaubt.

- **Fernwartung**
  Servicetechniker können sich per VPN in die Netzwerke ihrer Kunden verbinden, um dort aus der Ferne Wartungsarbeiten, Fehlerdiagnose, Updates usw. an Servern, Steuerungen und Maschinen durchzuführen.

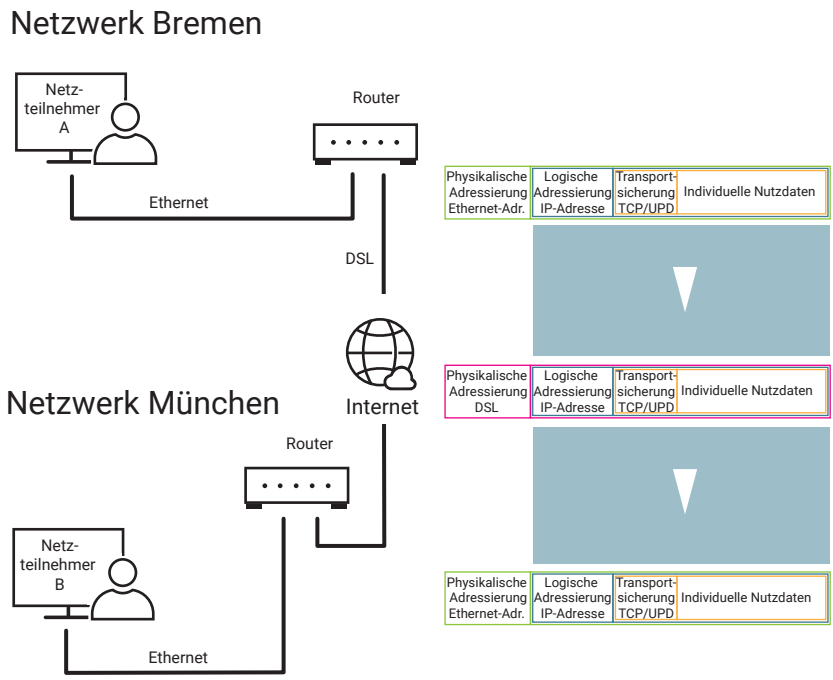
Anforderungen an ein VPN

Es geht also letztlich immer darum, Zugang zu einem entfernten Netzwerk zu bekommen.

Im Gegensatz zum normalen Routing muss VPN zusätzlich die für Datensicherheit typischen Anforderungen erfüllen:

- **Authentifizierung**
  Für den Zugriff auf den entfernten Netzwerkteil muss die Zugriffsberechtigung nachgewiesen werden.
- **Datenintegrität**
  Beim Empfang von Daten muss sichergestellt werden, dass diese auf dem Transportweg nicht verändert wurden.
- **Datensicherheit / Vertraulichkeit**
  Die übertragenen Daten müssen auf dem Transportweg vor Verfälschung oder Abhorchen durch unberechtigte Dritte geschützt sein.

Exkurs: normales Routing
Zur Erinnerung: Beim normalen Routing haben Quell- und Zielnetzwerk verschiedene Net-IDs. Die Net-IDs sind Bestandteil der IP-Adressen und dienen als Routing-Information. Die Adressen innerhalb des IP-Datenpaketes bleiben über die gesamte Strecke unverändert.



Die physikalischen Adressdaten hingegen wechseln von Teilabschnitt zu Teilabschnitt.

Innerhalb der lokalen Netzwerke erfolgen die Adressierung und der Datentransport über Ethernet, auf Internetebene über DSL und andere physikalische Übertragungsmethoden. Das IP-Paket bleibt dabei über die gesamte zu überbrückende Strecke unverändert.

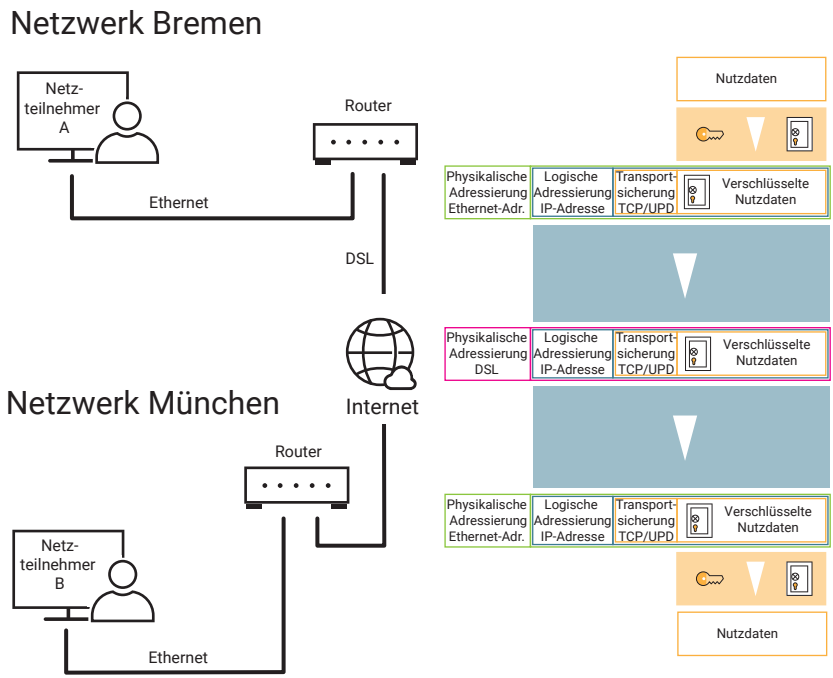
Datensicherheit
Solange die Übertragungswege durchgängig sind, gelangen die Daten auch mit herkömmlichem Routing zuverlässig von Netzwerk A zu Netzwerk B.

Ein Nachteil bei normaler Netzwerkkommunikation besteht allerdings darin, dass die Daten von jedem gelesen werden können, der physikalischen Zugang zu den Übertragungswegen hat. Nicht nur bei z.B. Bankdaten ist also ein erhebliches Sicherheitsrisiko gegeben.

Datenverschlüsselung
Wie bereits im vorangegangenen Kapitel kennengelernt, ist eine Möglichkeit, Daten vor Aushorchen oder Manipulation zu schützen, die Verschlüsselung.

Dritte, die den oder die verwendeten Schlüssel nicht kennen, können den verschlüsselten Datenstrom nicht ohne Weiteres lesen oder auswerten.

Dabei muss beiden Seiten der verwendete Schlüssel bekannt sein oder es muss eine asynchrone Verschlüsselung bzw. ein Hybridverfahren angewendet werden.



Lesbar sind allerdings die IP- und TCP-Adressierungsparameter. Dritte, die sich Zugriff auf fremde Daten oder ein fremdes Netzwerk verschaffen möchten, sehen anhand dieser Daten zumindest, wo das Zielnetzwerk im Inneren ggf. angreifbar ist.

VPN statt normalem Routing

Wie anfänglich bereits gesagt, können mittels VPN zwei Netzwerkteile an verschiedenen Standorten über das Internet bzw. ein öffentliches Netz verbunden werden.

Auch wenn VPN auf die ganz normalen IP-Netzwerkmechanismen aufsetzt, gibt es hinter den Kulissen ganz erhebliche Unterschiede.

VPN - Mögliche Topologien

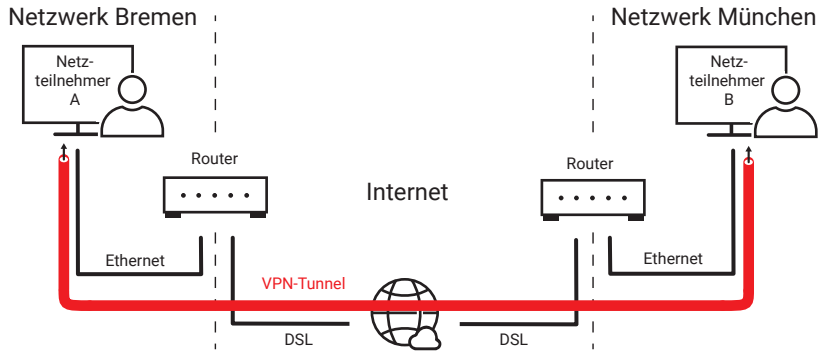
Es gibt drei grundlegende Topologien bei VPN-Lösungen:

- End-to-End
- Site-to-Site
- End-to-Site

Welche Variante zum Einsatz kommt, hängt letztlich davon ab, wie die VPN-Anbindung genutzt werden soll.

VPN - End-to-End

Bei End-to-End Lösungen werden zwei Netzwerkendgeräte über ein öffentliches Netz - z.B. das Internet - so miteinander verbunden, dass sie uneingeschränkt Netzwerkpakete miteinander austauschen können. Die Übertragungsstrecke durch das öffentliche Netz bezeichnet man auch als Tunnel, da der Datenverkehr zwischen den Endgeräten abgegrenzt zum restlichen Netzwerkverkehr abgewickelt wird.

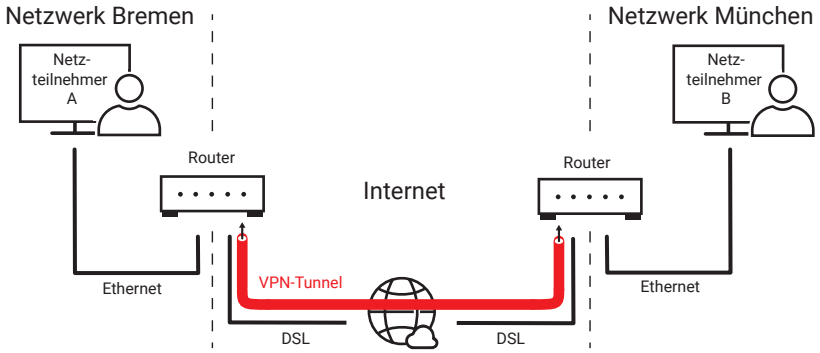


Ein Beispiel für eine End-to-End VPN-Anbindung ist ein Wartungszugang, über den ein Servicetechniker von seinem Büro aus z.B. Fehler an einem Kunden-System analysieren kann.

Damit das Ganze funktioniert, muss allerdings auf beiden PCs eine spezielle VPN-Software installiert sein. Ferner muss jeder PC speziell für den VPN-Zugriff konfiguriert werden.

VPN - Site-to-Site

Mit der VPN-Site-to-Site Technik werden zwei einzelne Netzwerke z.B. über das Internet miteinander verbunden.



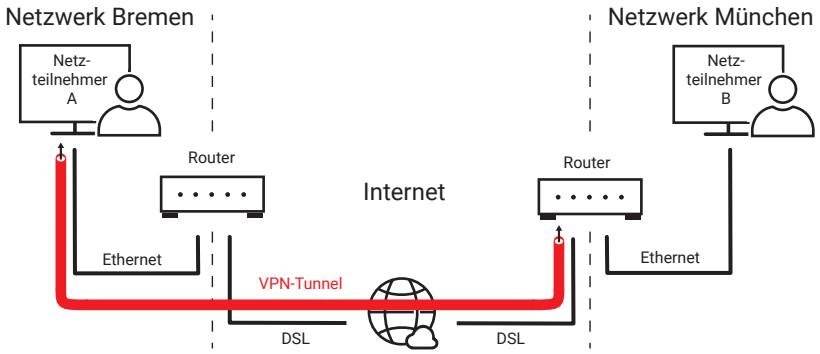
Der VPN-Tunnel wird zwischen zwei speziellen VPN-Routern aufgebaut. Die gesamte VPN-Konfiguration erfolgt in den Routern.

Die einzelnen Teilnehmer im Netzwerk benötigen keine spezielle Software und müssen auch nicht gesondert konfiguriert werden.

Site-to-Site Lösungen werden hauptsächlich zur Verbindung verschiedener Firmenstandorte eingesetzt.

VPN - End-to-Site

Die End-to-Site Lösung bietet einzelnen Endgeräten bzw. PCs Zugang zu einem gesamten Netzwerk am entfernten Standort.



Diese Lösung bietet sich für die Anbindung von Home-Office-Arbeitsplätzen an. Der Mitarbeiter kann von zu Hause die gesamte Infrastruktur des Firmennetzes nutzen.

VPN-Protokolle

Für die technische Umsetzung von VPN kommen in der Praxis mehrere Protokolle in Frage:

- PPTP - Point-to-Point Tunneling Protocol
- IPsec - IP Security Protocol
- L2TP - Layer 2 Tunneling Protocol
- OpenVPN
- WireGuard

Welches Protokoll zum Einsatz kommt, hängt von der VPN-Topologie und der verwendeten Hard- und Software ab.

PPTP - Point-to-Point Tunneling Protocol

Ursprünglich wurde PPTP von Microsoft und 3COM entwickelt, um abgesetzten PCs über eine Einwahlleitung Zugriff auf zentrale Server zu ermöglichen. Da PPTP von Hause aus in Windows-Betriebssystemen implementiert ist, erfreut es sich immer noch großer Verbreitung. Die Verschlüsselung von PPTP gilt allerdings nicht mehr als sicher.

Technische Grundlage von PPTP ist das PPP-Protokoll (siehe Kapitel: Übertragungsprotokolle), das unter anderem um eine Datenverschlüsselung und zusätzliche Authentifizierung erweitert wurde.

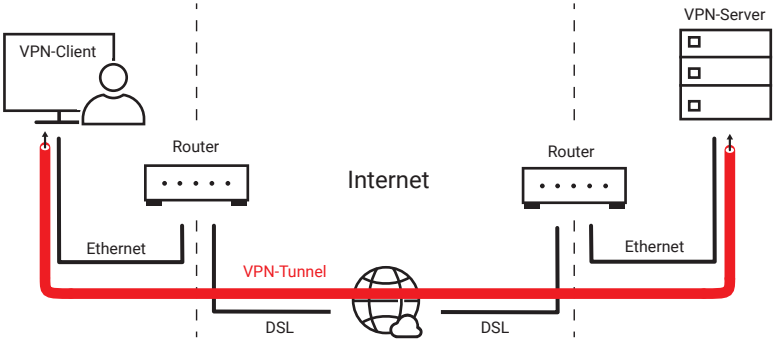
Durch die PPP-Implementierung hat PPTP den Vorteil, neben IP auch andere Protokolle wie z.B IPX (ehemals von Novell und Windows genutzt) übertragen zu können.

PPTP arbeitet zweistufig: Zunächst werden über eine Steuerverbindung auf TCP Port 1723 Authentifizierungs- und Schlüsseldaten ausgetauscht.

Anschließend werden eingekapselt in das GRE-Protokoll die PPP-Daten ausgetauscht. Die GRE-Kapselung (Generic Route Encapsulation) ist bildlich gesehen der Tunnel, durch den die PPP-Daten transportiert werden.

GRE hat den Charakter eines Transportprotokolls, das auf der gleichen Ebene wie z.B. TCP arbeitet und direkt in ein IP-Paket eingebettet wird.

PPTP arbeitet nach dem Client/Server-Verfahren. Der VPN-Client meldet sich also mit Aufbau der Steuerverbindung bei einem VPN-Server an. Es sind deshalb nur End-to-End VPN-Lösungen möglich.



IPsec - Internet Security Protocol

IPsec wurde speziell für die gesicherte Datenübertragung von IP-Datenverkehr über öffentliche Netze bzw. das Internet konzipiert. Als Security-Protokoll beinhaltet IPsec verschiedenste Funktionen, um die üblichen Anforderungen an Datensicherheit flexibel bedienen zu können.

SA - Security Association

Die Kommunikationspartner haben die Möglichkeit, bei den folgenden Punkten auszuhandeln, welche Verfahren bzw. welcher Standard genutzt werden sollen:

- Authentifizierung
- Datenintegrität
- Datensicherheit
- Schlüsselaustausch

Den daraus resultierenden Parametersatz bezeichnet man als Security Association.

SAD - Security Association Database

Festgehalten werden die Parametersätze in der Security Association Database, kurz SAD. So können, wenn VPN-Verbindungen zu mehr als einem Ziel betrieben werden,

in einer Art Liste verschiedene Parametersätze für jedes einzelne Ziel verwaltet werden. Um die oben genannten Sicherheitsmechanismen im Protokoll umzusetzen, bietet IPsec zwei Sicherheitsprotokollvarianten, die einzeln oder auch kombiniert genutzt werden können.

AH - Authentication Header

Mit Authentication Header, kurz AH, werden ausschließlich Authentifizierung und Datenintegrität abgesichert. Es ist also sichergestellt, dass man es mit dem gewünschten Kommunikationspartner zu tun hat und dass die Daten auf dem Übertragungsweg nicht verändert wurden. Die Übertragung erfolgt aber unverschlüsselt, so dass Dritte die Inhalte ggf. mitlesen können.

ESP - Encapsulating Security Protocol

Auch mit dem Encapsulating Security Protocol, kurz ESP, werden Authentifizierung und Datenintegrität abgesichert, aber zusätzlich werden die Daten verschlüsselt. Dazu werden natürlich auf beiden Seiten der VPN-Verbindung entsprechende Schlüssel benötigt.

IKE(v2) - Internet Key Exchange (Version 2)

Um beiden Seiten die benötigten Schlüssel und weitere Parameter zugänglich zu machen, nutzt IPsec das IKE-Protokoll. IKE nutzt dazu UDP-Port 500. Das ursprüngliche IKE ist inzwischen ersetzt durch IKEv2.

IKE arbeitet in zwei Phasen:

- **Phase 1**
  In der ersten Phase wird die Authentisierung abgesichert - es wird also sichergestellt, dass die Kommunikation mit dem tatsächlich gewünschten Partner stattfindet. Anschließend wird ein gesicherter Kommunikationskanal für die zweite Phase aufgebaut.

- **Phase 2**
  Im zweiten Schritt erfolgt über den gesicherten Kanal die Einigung über die genutzten Sicherheitsmechanismen.

Neben den verschiedenen Sicherheitsmechanismen gibt es auch noch zwei unterschiedliche Transportmodelle:

- **IPsec-Transportation**
  Der Datentransport erfolgt über normales Routing, wobei innerhalb des öffentlichen Netzes alle Daten bis auf die IP-Header gegen Fremdzugriff geschützt sind.

- **IPsec-Tunneling**
  Der durch VPN-Tunneling entstandene Netzwerkverbund stellt sich für die Netzwerkbenutzer so dar wie ein lokales Netzwerk. Der gesamte Datenstrom inklusive IP-Header ist geschützt.

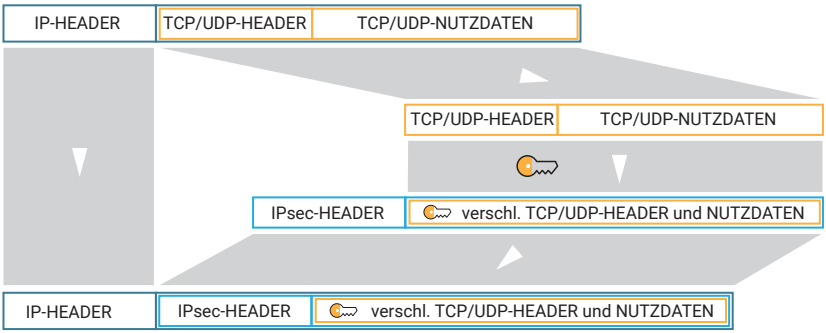
IPsec-Transportation

Der IPsec-Transportation-Modus wird bevorzugt bei End-to-End VPN-Lösungen eingesetzt. Damit das funktionieren kann, muss auf den beteiligten PCs eine Software installiert sein, welche das IPsec-Prozedere abwickelt.

Um die Daten gesichert über das öffentliche Netz zu bekommen, entnimmt der IPsec-Treiber den gesendeten TCP/IP-Paketen alle Inhalte, die protokolltechnisch oberhalb des IP-Teils liegen.

Der gesamte TCP- bzw. UDP-Teil - also Header und Nutzdaten - werden zusammen verschlüsselt und in einen IPsec-Rahmen verpackt.

Der IPsec-Rahmen wird dann in ein IP-Paket eingebaut, wobei die ursprünglichen IP-Adressinformationen erhalten bleiben.

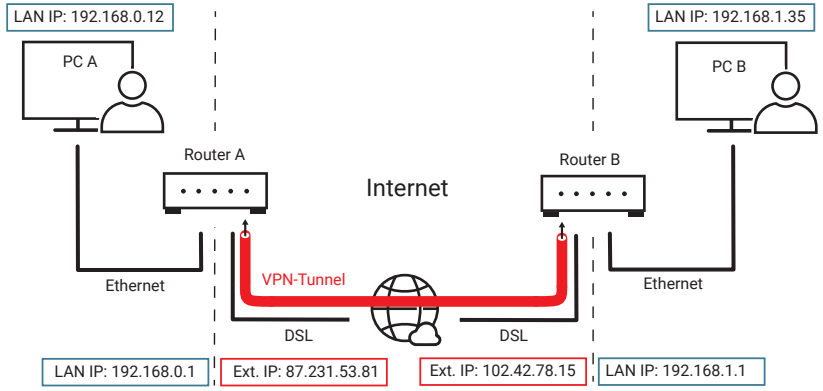


Der Datentransport wird also mit ganz normalem Routing bewerkstelligt, wobei die transportierten Daten und Port-Informationen geschützt sind.

IPsec-Tunneling

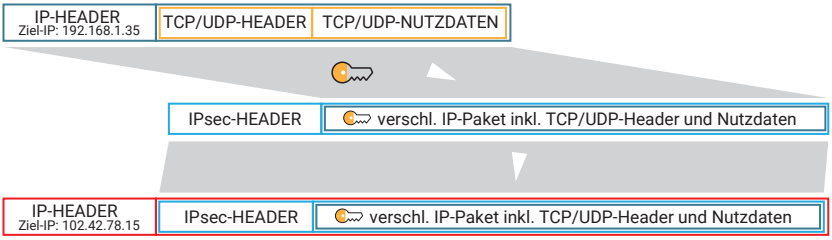
Wie anfänglich bereits gesagt, können mittels IPsec-Tunneling zwei Teilnetzwerke an verschiedenen Standorten über das Internet so verbunden werden, als würden Daten zwischen zwei lokalen Sub-Netzen ausgetauscht (Site-to-Site Lösung).

Die Abwicklung von IPsec übernehmen beim IPsec-Tunneling spezielle Router. Der Vorteil von IPsec-Tunneling gegenüber IPsec-Transportation liegt unter anderem in der Entlastung der beteiligten Endgeräte. Spezielle Treiber sind nicht nötig.



Sendet PC A ein Datenpaket an PC B, wird dieses zunächst von Router A entgegengenommen.

Router A verschlüsselt den gesamten IP-Paketanteil so, wie er ist, und verpackt ihn in einen IPsec-Rahmen. Der IPsec-Rahmen wird dann in ein neues IP-Datenpaket eingebaut, das an Router B adressiert ist.



Router B entschlüsselt das ursprüngliche IP-Paket und sendet es an PC B.

Für die PCs stellt sich die Datenübertragung so dar, als würde der Datenverkehr ganz normal geroutet.

Das Routing innerhalb des Internets erfolgt aber ausschließlich zwischen den beiden VPN-Routern.

Bei Bedarf kann so jedes Endgerät in Netzwerk A Daten mit jedem Endgerät in Netzwerk B austauschen; sicher und so, als wäre die Gegenstelle im selben lokalen Netzwerk.

Mit den verschiedenen zur Verfügung stehenden Transport- und Sicherheits-Modi ist IPsec höchst flexibel einsetzbar, bringt aber einen hohen Konfigurationsaufwand mit sich.

L2TP - Layer 2 Tunneling Protocol

L2TP ist ein reines Tunneling-Protokoll.

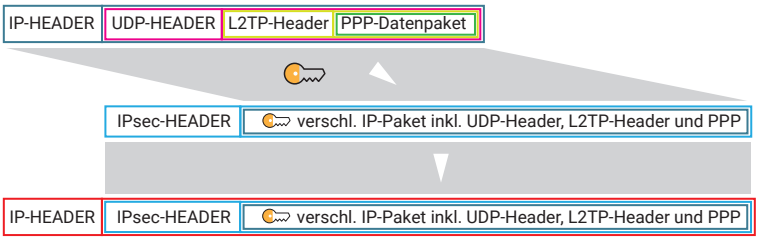
Zur Datenübertragung benutzt L2TP so wie auch PPTP das PPP-Protokoll. Die PPP-Daten werden mit einem L2TP-Header versehen und in ein UDP-Paket eingebettet.

L2TP übernimmt dabei folgende Aufgaben:

- Auf- und Abbau eines Datentunnels
- Kontrolle, ob Daten ihren Empfänger korrekt erreicht haben
- Nummerierung der Datenpakete, um beim Empfänger die Daten in die richtige Reihenfolge zu bringen

Allerdings arbeitet L2TP völlig unverschlüsselt. Unbefugte Dritte, die Zugang zu den Übertragungswegen haben, könnten alle Informationen ungehindert lesen. Damit ist L2TP allein für die Realisierung eines VPN-Tunnels nicht geeignet.

Um die nötige Sicherheit zu gewinnen, wird L2TP meist zusammen mit IPsec eingesetzt.



Nun könnte man natürlich fragen: Wenn L2TP allein ohnehin nicht sicher ist, warum nicht gleich IPsec nutzen?

- Zum einen gibt es Anwendungen, bei denen innerhalb eines vertraulichen Netzwerkes Daten getunnelt werden sollen - hier bietet L2PT ja alles, was benötigt wird.
- Zum anderen kann IPsec nur IP-Pakete tunneln. L2TP hingegen kann wegen des verwendeten PPP-Protokolls auch andere Pakettypen transportieren - mittels IPsec auch verschlüsselt.

OpenVPN

OpenVPN ist als freie Software nach GNU GPL (General Public Licence) lizenziert.

Die Funktionsvielfalt und die Fülle an Konfigurationsmöglichkeiten von OpenVPN ist sehr groß, so dass wir hier nur die Eigenschaften und Möglichkeiten zusammenfassen wollen:

- OpenVPN basiert auf der OpenSSL-Bibliothek und kann dadurch alle Verschlüsselungs-, Authentifizierungs- und Zertifizierungsmöglichkeiten von SSL/TLS nutzen.
- Es kann wahlweise UDP oder TCP als Basisprotokoll verwendet werden. Bei Verwendung von TCP erfolgt die Kommunikation über Port 443, also den HTTPS-Port - damit kann OpenVPN die meisten Firewalls ohne Probleme überwinden.
- Es ist lauffähig auf nahezu allen bekannten Betriebssystemen und vielen standardisierten Hardware-Plattformen (und damit neben PCs und Servern auch in Routern, Embedded Systemen und Smartphones nutzbar).
- Es ist mit IPsec kombinierbar.
- Alle drei VPN-Topologien (End-to-End, Site-to-Site und End-to-Site) werden unterstützt.
- Selbst sehr große Netzverbünde mit mehr als 1.000 abgesetzten Zugängen können mit OpenVPN aufgebaut werden.

Eine Besonderheit von OpenVPN liegt darin, dass neben normalem Routing auch Bridging unterstützt wird.

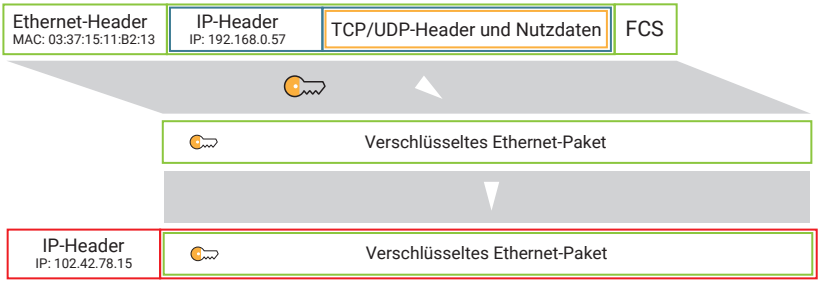
Routing

Zur Erinnerung: Beim Routing wird anhand von IP-Adresse und Subnet-Mask ermittelt, ob die Kommunikation im lokalen Netzwerk erfolgt oder über welchen Weg die Daten in welches Zielnetzwerk weiter übertragen werden. Das funktioniert natürlich

nur auf der Ebene von IP.

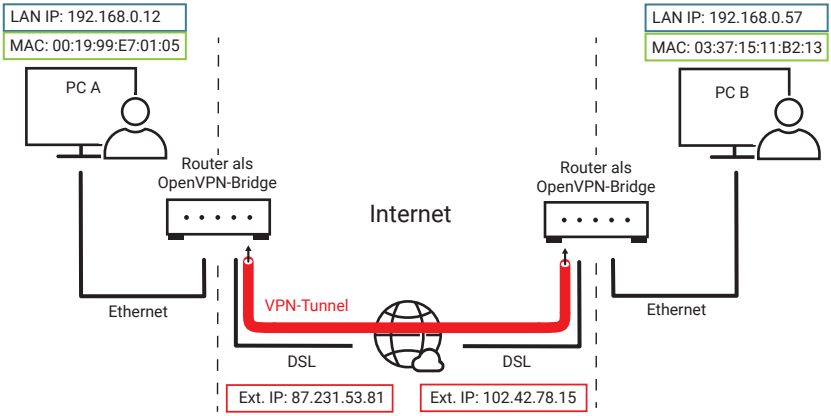
Bridging

Beim Bridging wird das vollständige Ethernet-Datenpaket von einem Teilnetz ins andere übertragen. Dazu wird das gesamte Paket zunächst verschlüsselt und dann in ein IP-Paket eingebettet, das an die externe IP-Adresse der zweiten Bridge adressiert ist.



Bei IP-Datenverkehr hat das den Vorteil, dass beide Teilnetze im selben IP-Adressbereich liegen können.

Für den einzelnen Netzwerkteilnehmer stellt sich der Verbund der Teilnetze wie ein einzelnes lokales Netzwerk dar.



Darüber hinaus ist die Übertragung nicht ausschließlich auf IP-Datenpakete beschränkt. Auch andere Protokolle wie z.B. IPX können transportiert werden.

Auch wenn es zunächst paradox klingt - OpenVPN gilt als besonders sicher, weil alle Quelltexte frei zugänglich sind. Damit ist sichergestellt, dass Programmierfehler oder versteckte Hintertüren sehr schnell erkannt werden können.

WireGuard

WireGuard ist die jüngste unter den VPN-Techniken und ebenfalls ein Open-Source-Projekt.

Bei der Entwicklung von WireGuard wurden die folgenden Kriterien in den Vordergrund gestellt:

- sorgfältig geplantes Konzept
- einfache Einrichtung und Handhabung
- hohe Performance
- Verwendung aktueller Verschlüsselungs- und Sicherheitstechniken
- geringe Menge an Quellcode

Zur Übertragung nutzt WireGuard UDP, wobei der Port frei wählbar ist.

Zum Zeitpunkt der Drucklegung dieser Ausgabe war die Entwicklung von WireGuard noch nicht abgeschlossen. Weitere Details zum WireGuard-Projekt finden Sie unter https://www.wireguard.com.

Der Weg ins Internet

Eine entscheidende Einschränkung der heute üblichen Ethernet-Technik ist die maximale Distanz von 100m. Zwar können mit Hilfe entsprechender Komponenten wie Hubs, Switches und Routern auch größere Entfernungen erreicht werden, aber auch damit ist die Ausdehnung eines Ethernet-Netzwerkes auf das Grundstück einer Firma bzw. die Wohnung eines privaten Nutzers begrenzt.

Geht es z.B. darum, eine Verbindung mit dem Internet herzustellen (Datenfernübertragung, kurz DFÜ), sind oft mehrere Kilometer zu überbrücken. Internetzugänge werden deshalb bis auf wenige Ausnahmen über das öffentliche Telefon-, Kabelfernseh- oder Mobilfunknetz angebunden. In seltenen Fällen werden auch Satellitenrichtfunkstrecken zur Verbindung mit dem Internet genutzt.

Ursprüngliche Internetzugänge

Insbesondere das Telefonnetz hat sich in den letzten Jahren technisch sehr verändert. Ursprünglich ausschließlich für die Übertragung von Sprache ausgelegt, wurden ab Ende der 1990er Jahre immer mehr Datendienste über die vorhandene Technik und Infrastruktur mitübertragen. Beide Dienste sind inzwischen als ALL-IP-Anschlüsse zusammengewachsen und nutzen als Basistechnik TCP/IP.

Dadurch verlieren einige Zugangstechniken wie analoge Modems und ISDN ihre Relevanz. Da diese Techniken aber ein Basiswissen zur auch heute noch benötigten Übertragungstechnik vermitteln, wollen wir trotzdem kurz darauf eingehen.

Analoge Modems

Modem steht für Modulator-Demodulator. Der Zugang über analoge Modems ist die ursprüngliche Art des Internetzugangs und wird heute zumindest für Zugänge ans öffentliche Netz nicht mehr verwendet. Bei Inhouse-Anwendungen z.B. zum Überbrücken größerer Distanzen auf einem Firmengelände kommen analoge Modems aber noch zum Einsatz.

Zwischen das Endgerät, meist ein PC, und den Telefonanschluss bzw. eine Telefonleitung, wird ein Modem geschaltet. Als Schnittstelle zwischen z.B. PC und Modem wird meist die serielle Schnittstelle (COM-Port / RS232) oder USB verwendet. Alternativ zu den externen Modems gibt es PC-Einsteckkarten, welche die Modem-Funktionen innerhalb des PC abwickeln.

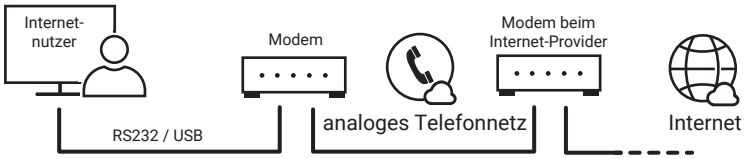
Wurde für die Übertragung das öffentliche Telefonnetz benutzt, musste zunächst eine Einwahlverbindung zum Internet-Provider hergestellt werden. Auch diese Aufgabe übernahm das Modem.

Für die Übertragung wurden die digitalen Informationen auf eine Trägerfrequenz aufmoduliert. Wir wollen an dieser Stelle nicht näher auf die angewendeten Modulationsverfahren eingehen, sondern nur eine beispielhafte Erklärung dieser Technik geben.

Die Trägerfrequenz kann man sich vorstellen wie einen bestimmten hörbaren Ton aus dem Frequenzbereich der Sprache (300 Hz – 3.400 Hz).

Der zu übertragende Datenstrom wird in einige Bits große Blöcke zerteilt. Je nachdem welches Bitmuster vorliegt, wird der Ton in einer für dieses Bitmuster vorgegebenen Art verändert.

Am anderen Ende der Verbindungsstrecke übernimmt ein zweites Modem die umgekehrte Aufgabe (Demodulation). Aus den empfangenen Tönen wird wieder ein Datenstrom zurückgewonnen.



Ursprünglicher Internetzugang über Telefonnetz und Modem

Durch den eingeschränkten Frequenzbereich von analogen Telefonanschlüssen bzw. den verlegten Leitungen liegt die maximale Datenübertragungsrate bei 33kBit/s vom Teilnehmer in Richtung Vermittlungsstelle (Upstream). Von der Vermittlungsstelle in Richtung Teilnehmer (Downstream) sind maximal 56kBit/s möglich.

Ein großer Nachteil von DFÜ über analoge Telefonanschlüsse war neben der geringen Übertragungsrate die Tatsache, dass parallel zu DFÜ nicht telefoniert werden konnte.

ISDN - Integrated Services Digital Network

Die Einführung der All-IP-Anschlüsse hat die ISDN-Technik abgelöst, so dass es ISDN im deutschen Telefonnetz nicht mehr gibt.

Der wesentliche Unterschied von ISDN zum analogen Telefonanschluss bestand darin, dass bei ISDN selbst analoge Sprachdaten bereits am Standort des Teilnehmers in digitale vermittlungstechnische Daten umgewandelt wurden.

Vom Teilnehmer zur Vermittlungsstelle wurden also ausschließlich digitale Daten in Form von ISDN-Netzwerkpaketen ausgetauscht.

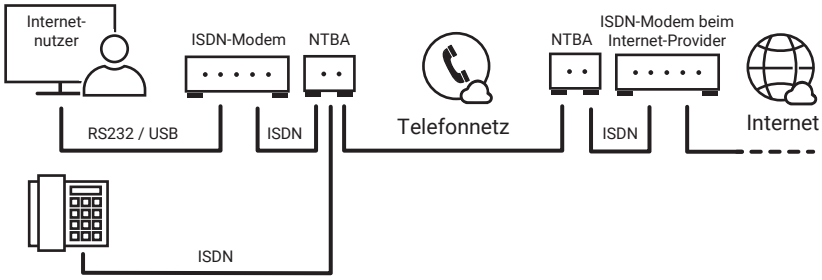
ISDN steht für Integrated Services Digital Network, was locker übersetzt so viel bedeutet wie: Integriertes digitales Netzwerk für verschiedene Dienste.

Neben der Übertragung von Sprache erlaubte ISDN von Hause aus den Austausch digitaler Daten z.B. für Fax und DFÜ.

Eine Modulation von DFÜ-Daten war bei ISDN im eigentlichen Sinne nicht nötig. Stattdessen wurden die zu übertragenden Daten in ISDN-Pakete verpackt und versendet, wobei auch hier zunächst eine Wählverbindung nötig war.

Trotzdem bezeichnete man die externen ISDN <-> DFÜ-Datenumsetzer gemeinhin als ISDN-Modems.

Zwischen ISDN-Modem und dem Telefonnetz bereitete der NTBA (Network Termination for ISDN Basic rate Access) die ISDN-Daten physikalisch so auf, dass sie zur Vermittlungsstelle übertragen werden konnten. Die Schnittstelle zwischen den ISDN-Endgeräten und dem NTBA wurde als S0-Bus bezeichnet.



ISDN stellte dem Teilnehmer zwei Kanäle (Bereiche im ISDN-Paket) zur Verfügung, die auch für unterschiedliche Dienste, z.B. Telefonieren und DFÜ, genutzt werden konnten.

Pro Kanal wurden 64kBit/s übertragen. Bei paralleler Nutzung beider Kanäle (Kanalbündelung) erhöhte sich die Transferrate auf 128kBit/s.

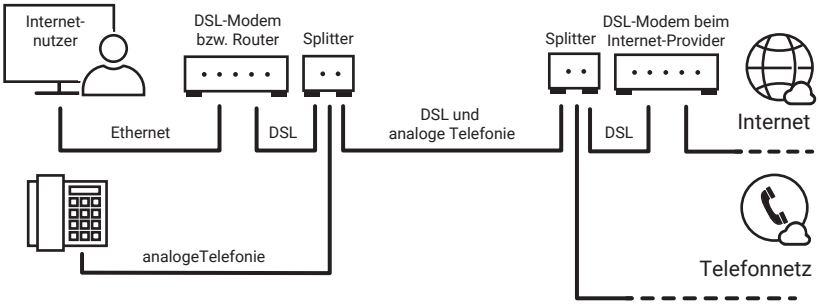
Aktuelle Internetzugänge

DSL - Digital Subscriber Line

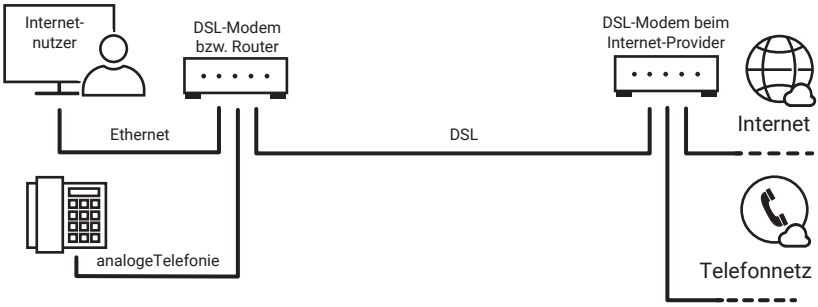
Digital Subscriber Line (deutsch: Digitale Teilnehmeranschlussleitung) war lange Zeit die attraktivste Möglichkeit, sich mit dem Internet zu verbinden.

Analoge Anschlüsse arbeiten auf dem Kabel mit Frequenzen bis max. 3,5 kHz. Bei ISDN liegt die Obergrenze bei ca. 40 kHz. DSL nutzt ausschließlich Frequenzen, die oberhalb 40kHz bis ca. 1 MHz angesiedelt sind.

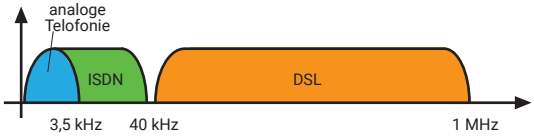
Damit konnte DSL parallel zu analogen oder ISDN-Anschlüssen über dasselbe Kabel betrieben werden. Am Standort des Teilnehmeranschlusses wurde über einen Splitter (eine Frequenzweiche) das DSL-Signal von den Telefonsignalen getrennt.



Bei der heute üblichen All-IP-Technik werden Telefonie und Daten erst im Router separiert.



Die Übertragung der DSL-Daten funktioniert ähnlich wie beim analogen Modem, nur dass gleichzeitig mit mehreren, deutlich höheren Trägerfrequenzen gearbeitet wird.



DSL gibt es in verschiedenen Varianten:

ADSL - Asymmetric Digital Subscriber Line
ADSL-Zugänge werden meist von Privatkunden genutzt und lassen beim Download Übertragungsraten von max. 25MBit/s zu. Da die Upstream-Geschwindigkeit nur ca. ein Achtel der Downstream-Geschwindigkeit beträgt, spricht man auch von asymmetrischer Datenübertragung.

SDSL- Symmetric Digital Subscriber Line
Bei SDSL wird in beide Richtungen mit der gleichen Übertragungsgeschwindigkeit von max. 4MBit/s gearbeitet. SDSL-Zugänge werden bevorzugt von gewerblichen Kunden genutzt - z.B. um zwei Firmenstandorte netzwerktechnisch miteinander zu verbinden.

VDSL - Very Highspeed Digital Subscriber Line
Da immer mehr Dienste wie z.B. Fernsehen oder Telefonie das Internet als Übertragungsweg nutzen, steigt zunehmend der Bedarf an sehr schnellen Internetzugängen. VDSL arbeitet ähnlich wie SDSL, aber mit deutlich höheren Übertragungsraten von über 200MBit/s.

Für alle DSL-Standards gilt: Je größer die Entfernung zur Vermittlungsstelle, desto geringer die mögliche Übertragungsgeschwindigkeit.

Auf Grund der hohen Übertragungsgeschwindigkeit tauschen DSL-Modems die Daten mit dem PC direkt über Ethernet aus. Eine häufige Variante ist ein Ethernet-Router mit integriertem DSL-Modem.

Kabel-Modem

Der Internetzugang über ein Kabel-Modem ist inzwischen eine echte Alternative zum DSL-Anschluss. Der Zugang erfolgt über das Kabelfernsehnetz. In den 80er Jahren

wurde das Kabelfernsehnetz für die Verteilung von Fernseh - und Radiokanälen aufgebaut und war nur dazu bestimmt, Signale vom Anbieter zum Kunden zu transportieren. Nachdem die Netzbetreiber den zunehmenden Bedarf an Internetzugängen erkannten, wurde der notwendige Rückkanal vom Kunden Richtung Anbieter nachgerüstet.

Physikalisch sind die Bestandsnetze mit Koaxialkabeln aufgebaut. Für Netzerweiterungen und neue Netze kommen Lichtwellenleiter zum Einsatz.

Die Verbindung zwischen Kabelfernsehnetz und lokalem Netzwerk bildet das Kabel-Modem bzw. ein speziell dazu ausgestatteter Router.

Die mögliche Übertragungsgeschwindigkeit liegt bei 32MBit/s und mehr.

Internetzugang über Mobilfunk

Eine Alternative zu den vorangegangenen Festnetzvarianten ist die Verbindung mit dem Internet über die Mobilfunknetze.

Eine detaillierte Beschreibung der Mobilfunktechnik würde den Rahmen des Buches sprengen. Deshalb wollen wir an dieser Stelle nur einen sehr oberflächlichen Überblick vermitteln.

Die Mobilfunkstandards gehen gerade in die fünfte Generation und obwohl sich das 5G-Netz in Deutschland gerade im Aufbau befindet, ist 5G bereits in aller Munde.

Hier zunächst einmal ein Überblick über die Mobilfunkgenerationen:

- 1G
  Die erste Generation begann 1958 mit dem A-Netz. Es ging fast ausschließlich um Telefonie im Auto. Obwohl die benötigte Technik fast kofferraumfüllend war, konnte der Teilnehmer nicht selber wählen. Verbindungen wurden handvermittelt über das analoge A-Netz abgewickelt. Datenübertragung war zu dieser Zeit noch kein Thema. Ab 1972 konnten die Teilnehmer im B-Netz selber wählen.1986 wurde dann das C-Netz aufgebaut und die ersten portablen Mobiltelefone waren verfügbar.

- 2G
  Mit dem D-Netz wurde 1992 von analoger Sprachübertragung auf Digitaltechnik gewechselt. Ein Jahr danach folgte der Aufbau des E-Netzes. Beide Netze benutzen GSM (Global System for Mobile Communication) als technischen Standard

und erlauben nun auch die Übertragung von Daten - zunächst als SMS-Textnachrichten und mit 9,6 Kbits/s sehr langsam.

- 2.5G
  GPRS (General Packet Radio Service) machte 2001 die Mobilfunkwelt internetfähig. Mit max. 54 Kbits/s allerdings immer noch sehr langsam.

- 2.75G
  Mit EDGE (Enhanced Data Rates for GSM Evolution) wurde 2006 durch neue Modulations- und Kompressionsverfahren die mögliche Übertragungsgeschwindigkeit im GSM-Netz nochmal auf 150 Kbits/s erhöht.

- 3G
  Als neuer Mobilfunkstandard kam 2004 UMTS (Universal Mobile Telecommunications System), was den Aufbau eines neuen, engmaschigeren Netzes und neue Endgeräte erforderte. Durch die deutlich höhere Bandbreite (ca. 380 Kbits/s) von UMTS begann auch der Siegeszug der mobilen Internetnutzung.

- 3.5G
  2006 brachte HSPA (High Speed Downlink Packet Access) als Erweiterung von UMTS noch mal einen deutlichen Geschwindigkeitsschub auf bis zu 42 Mbits/s.

- 4G
  2010 wurde mit LTE (Long Term Evolution) der zur Zeit noch aktuelle Mobilfunkstandard eingeführt. LTE setzt auf die Technik von UMTS auf. Während anfänglich Übertragungsraten bis zu 50Mbits/s möglich waren, werden durch Frequenzbündelung und andere Techniken (LTE Advanced) Geschwindigkeiten von bis zu 400-500 Mbits/s erreicht.

- 5G
  Wie bereits gesagt ist das 5G-Netz zur Zeit im Aufbau und soll 2020 zumindest in Teilen in Betrieb gehen. Versprochen werden Übertragungsraten von bis zu 20 Gigabits/s. Fast wichtiger sind aber die sehr niedrigen Latenzzeiten (Reaktionszeiten der Netzwerkdienste), die 5G mitbringen soll. Damit kommt 5G nah an ein Echtzeitverhalten, wie es zum Beispiel für Industrieanwendungen und autonomes Autofahren benötigt wird.

Hier noch mal eine Übersicht der verschiedenen Standards:

Generation	Einführung	Standard	max. Datenrate
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

Die angegebenen Geschwindigkeiten sind maximale Übertragungsraten. Welche Geschwindigkeit wirklich erreicht wird, hängt unter anderem davon ab, wie weit die Entfernung zum nächsten Funkmast ist und wieviele Teilnehmer in der gleichen Funkzelle Daten austauschen.

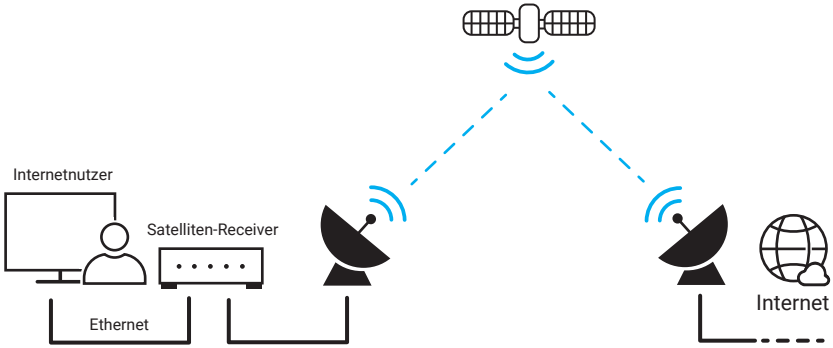
Weitere Erklärungen zur Technik der beschriebenen Mobilfunkstandards finden Sie im Netzwerk-ABC.

Technische Voraussetzungen
Um sich als User über das Mobilfunknetz mit dem Internet zu verbinden, benötigt man entweder ein Smartphone bzw. Tablet, einen Surf-USB-Stick ober einen Mobilfunk-Router.

Um beliebige Ethernet-Endgeräte über Mobilfunk mit dem Internet zu verbinden, können Router mit dem entsprechenden Mobilfunkzugang eingesetzt werden.

Internetzugang über Satellit

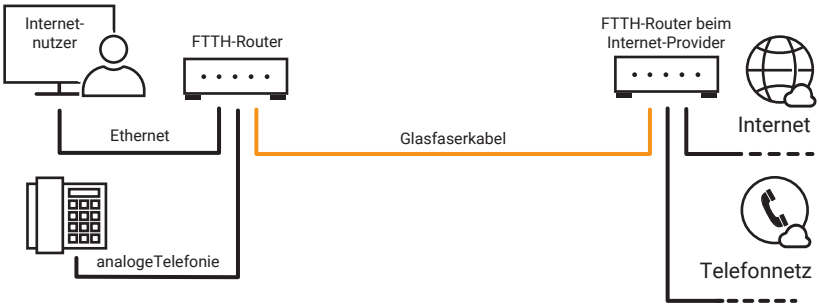
Für sehr entlegene Standorte gibt es die Möglichkeit, den Internetzugang über eine Satellitenanbindung zu bekommen. Der Hardware-Aufwand für einen solchen Anschluss ist deutlich höher als bei den sonst üblichen Zugängen, da eine Parabolantenne und ein spezieller Satelliten-Receiver benötigt werden.



Internetanbindungen über Satellit erlauben Übertragungsraten von bis zu 30Mbits/s im Downstream, also Datenverkehr zum Nutzer. Die Upstream-Geschwindigkeit ist deutlich langsamer. Bei IoT-Anwendungen sollte man im Auge behalten, dass die Latenzzeiten deutlich höher sind als bei allen anderen Internetanbindungen.

FTTH - Fibre to the Home

Inzwischen gibt es Internetanbieter, die eine Glasfaseranbindung bis zum Hausübergabepunkt bzw. bis in die Wohnung anbieten. FTTH erlaubt z.Zt. bis zu 1.000MBit/s und benötigt einen Router mit speziellem FTTH-Anschluss. Das FTTH-Netz ist im Moment noch im Aufbau und nur an wenigen Standorten verfügbar.



Der Browser als Bedienoberfläche

Vorweg sei gesagt, dass dieses Kapitel kein Tutorial für das Erstellen von Webseiten bzw. Web-Anwendungen sein soll - die aufgeführten Beispiele geben nur eine Übersicht über die Möglichkeiten und technischen Hintergründe.

In den ersten 20 Jahren seines Daseins war die Nutzung des Internets für den gewöhnlichen Anwender kaum interessant. Eine für heutige Verhältnisse kleine Gruppe von Insidern musste kryptische Befehlszeilen eintippen, um Informationen austauschen zu können.

Heute ist ein Leben ohne Internet kaum noch denkbar. Bankgeschäfte, Onlinebestellungen, Urlaubsbuchungen bis hin zur Partnersuche - fast alles kann man heute über das Internet erledigen. Im Jahr 2019 verbrachte jeder Erwachsene in Deutschland durchschnittlich drei Stunden täglich Zeit mit dem Internet - bei Jugendlichen waren sogar sechs Stunden.

Mit Smartphone, Tablet, Notebook und vernetzten Autos hat man das Internet quasi ständig bei sich.

Der Durchbruch für diesen Internet-Siegeszug kam mit Einführung des Browsers als für den normalen Nutzer handhabbares Visualisierungs-Tool.

WWW - World Wide Web

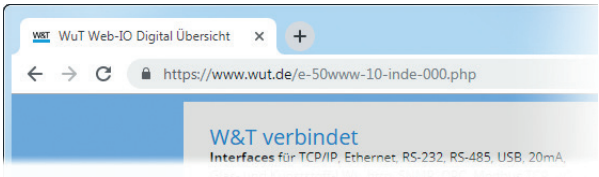
Jeder nutzt heute wie selbstverständlich den Browser, ohne sich Gedanken darüber zu machen, welche Technik dahintersteckt.

Bereits 1994 wurde das World Wide Web Consortium, kurz W3C, gegründet - eine Organisation, die es sich zur Aufgabe gemacht hat, weltweit einheitliche Standards für Web-Techniken zu schaffen.

Noch mal zur Erinnerung: Der Browser ist eine Client-Anwendung und baut bei Bedarf eine Verbindung zum gewünschten Webserver auf. Die Übertragung der Daten wird über das HTTP- bzw. HTTPS-Protokoll abgewickelt (Details hierzu im Kapitel Web-Protokolle).

URL - Uniform Resource Locator

Die URL ist die Adressangabe, die man im Browser einträgt und die vorgibt, wo der gewünschte Inhalt abgerufen wird.



Neben der eigentlichen Adresse des Webservers enthält die URL noch weitere Angaben und Parameter:

```
protokoll://hostname [:tcp-port] [/pfadname][/filename][?weitere parameter]
```

`protokoll`

Für den Aufruf einer Webseite normalerweise HTTP oder HTTPS

In der Vergangenheit wurden je nach Browser und Betriebssystem auch die Protokolle FTP oder TELNET unterstützt. Bei der Angabe des Protokolls gibt es keinen Unterschied zwischen Groß- und Kleinschreibung.

`hostname`

Domainname des Servers oder IP-Adresse. Auch hier wird nicht zwischen Groß- und Kleinschreibung unterschieden.

`:tcp-port`

Die Angabe des TCP-Ports ist optional und muss nur erfolgen, wenn von den Standardports für HTTP (80) oder HTTPS (443) abgewichen wird.

`/pfadname`

Ähnlich wie beim PC hat auch ein Webserver eine Verzeichnisstruktur. Befindet sich der gewünschte Inhalt in einem Unterverzeichnis, wird der dahin führende Pfad mit vorangestelltem Slash angegeben. Bei der Pfadangabe ist die Groß-/Kleinschreibung zu beachten.

`/filename`

Hier kann der Name der aufzurufenden Datei angegeben werden. Wenn kein Dateiname angegeben wird, verwendet der Webserver die Datei „index.html“ oder „index.php“. Auch beim Dateinamen ist die Groß-/Kleinschreibung zu beachten.

`?weitere parameter`

Mit einem Fragezeichen getrennt können innerhalb der URL auch weitere Parameter angegeben werden. Mehrere Parameter werden mit „&“ voneinander getrennt. Ob Groß-/Kleinschreibung relevant ist, hängt von der Programmierung der aufgerufenen Webseite ab.

Beispiel - URL für die Datenschutzrichtlinien von Wiesemann & Theis.:

```
https://www.wut.de/e-wwwww-ds-rdde-000.php?Reference=datenschutz
```

HTML – Hypertext Markup Language

Der Browser bekommt vom Webserver nach Aufruf der URL die zugehörige Webseite als HTML-Datei übergeben. Im HTML-Format wird dem Browser übermittelt, welche Inhalte wie angezeigt werden sollen.

HTML-Dateien haben die Namensendung .htm oder .html.

HTML ist eine Auszeichnungssprache (Markup Language), die sich aus Schlüsselwörtern – auch Tags genannt – und den darzustellenden Inhalten zusammensetzt. Die Tags geben an, in welcher Art und Weise der nachfolgende Text darzustellen ist. So lassen sich z.B. Schriftgröße, -art und -ausrichtung vorgeben. Inhalte können in Tabellen oder in Form einer numerischen Aufzählung dargestellt werden, die Farbe von Text und Hintergrund kann festgelegt werden usw. Der Browser interpretiert diese Angaben und stellt sie entsprechend dar.

HTML-Tags

Für HTML-Tags gilt ein festes Schema:

- Einzelne Tags sind von spitzen Klammern umschlossen
  `<HTML-Tag>`.
- Das eigentliche Tag kann durch Angabe von Attributen erweitert werden.
  `<HTML-Tag Attribut="xy">`
- Überwiegend wird durch die paarweise Verwendung von Tags der Anfang und das Ende ihres Gültigkeitsbereichs festgelegt; die definierten Eigenschaften gelten dann für alles, was zwischen den Tags steht. Das schließende Tag wiederholt das öffnende Tag mit einem vorangestellten Schrägstrich.
  Beispiel: `<title>Willkommen</title>`
- Bei HTML-Tags wird nicht zwischen Groß- und Kleinschreibung unterschieden. `<HTML>` ist gleichbedeutend mit `<html>`.

Grundsätzlicher Aufbau einer HTML-Datei

In der ersten Zeile wird mit dem `<!DOCTYPE ...>`-Tag gekennzeichnet, dass es sich um eine HTML-Datei handelt.

Der eigentliche Inhalt wird mit `<html>` eingeleitet und endet mit `</html>`. Man unterscheidet beim weiteren Aufbau einer Seite zwischen Kopf und Körper.

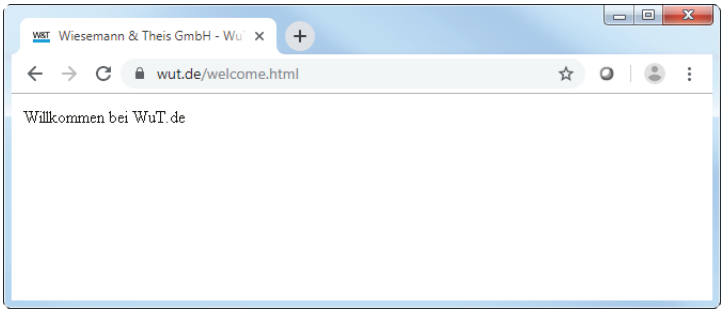
Alle Angaben im Kopf bleiben für den Betrachter unsichtbar und enthalten Eigenschaften der Seite, die nicht direkt die Darstellung betreffen. Einzige Ausnahme ist der Titel, der in der Titelleiste des Browserfensters angezeigt wird. Die Kopfinformationen stehen zwischen den Tags `<head>` und `</head>`.

Auf den Kopf folgt der Seitenkörper, der mit dem `<body>`-Tag eingeleitet wird. Im Körper der HTML-Seite sind alle Angaben zu finden, die den eigentlichen Inhalt der Seite und dessen Darstellung betreffen. Das Ende des Körpers wird mit dem `</body>` Tag gekennzeichnet.

Hier ein einfaches Beispiel:

```
<!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>
```

Im Browser sieht das dann so aus:



Neben Texten können mit Hilfe von HTML auch Grafiken eingebunden werden.

Sogar multimediale Inhalte wie Musik, Sprache oder Filmsequenzen lassen sich per HTML einbinden. Das HTML-Dokument selbst transportiert dabei ausschließlich Textinhalte. Für jedes andere darzustellende Element wird via HTML angegeben, von wo es geladen werden kann, wo es auf dem Bildschirm erscheinen und in welcher Größe es dargestellt werden soll.

Eine gute Übersicht zu HTML und Erklärungen zu allen verfügbaren Tags finden Sie unter https://wiki.selfhtml.org.

Hyperlinks

Die wohl wichtigste Eigenschaft, die HTML mitbringt, ist die Verwendung von Hyperlinks. Texte und andere Elemente einer Webseite können mit einem Hyperlink, also einem URL-Verweis auf eine andere Webseite, versehen werden. Klickt der Anwender auf ein solches verlinktes Element, wird er auf die verlinkte Webseite weitergeleitet.

Wir erweitern den HTML-Code um einen Hyperlink:

```
<body>
  Willkommen bei <a href="https://www.wut.de">WuT.de</a>
</body>
```

Bei einem Mausklick auf „WuT.de“ werden wir nun auf die Homepage von W&T gelenkt.

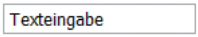
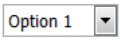
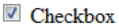
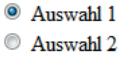
Das Pfadattribut des Tags <a href=“Pfadangabe“> kann die Pfadangabe entweder in absoluter oder in relativer Form enthalten.

- Absolut: Es wird die komplette URL angegeben, auf die der Hyperlink verweisen soll.
- Relativ: Es wird nur der Name der Datei angegeben auf die zugegriffen werden soll. Die Datei wird dann im gleichen Verzeichnis gesucht, in dem sich auch die aktuelle HTML-Datei befindet.

Formulare

Damit der Anwender bei Bedarf auch Informationen an den Webserver senden kann, gibt es in HTML die Möglichkeit, Formulare einzubinden.

Formulare beinhalten Elemente, die dem Anwender z.B. die Möglichkeit geben, Text einzugeben oder in einer Auswahl etwas zu markieren. Hier die wichtigsten Formularelemente:

Textfelder	
Textauswahlboxen	
Checkboxen	
Radiobuttons	
Durch Anklicken eines „Submit-Buttons“ werden die Formularinhalte vom Browser zum Webserver gesendet.	

Webseiten, die in reinem HTML aufgebaut sind, haben einen entscheidenden Nachteil - einmal im Browser geladen erfolgt keine Aktualisierung ohne Eingriff des Anwenders.

Webseiten werden deshalb heutzutage dynamisch und nicht mehr in reinem HTML aufgebaut.

Dynamische Webseiten

Die Anforderungen an Webseiten haben sich in den letzten Jahren so verändert, dass HTML als formatgebende Auszeichnungssprache nicht ausreicht, um diesen gerecht zu werden.

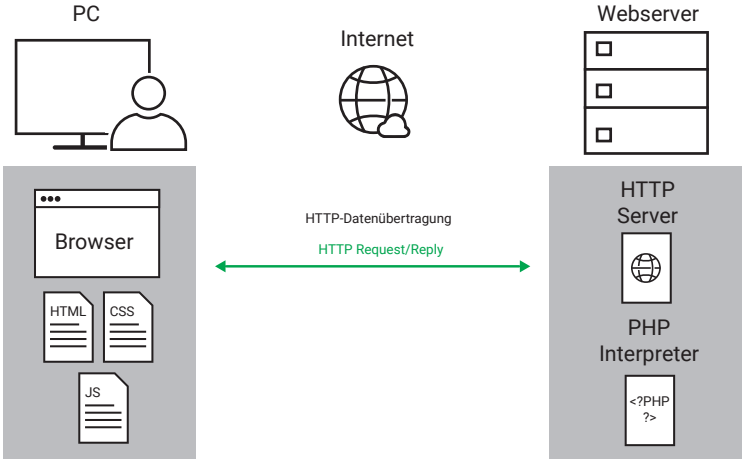
Die Anbindung an soziale Medien, Online-Shopping bis hin zu technischen Anwendungen wie Smart Home und IoT verlangen einen stetigen Datenaustausch zwischen Browser und Server und eine permanente Aktualisierung der Anzeige.

Um das zu leisten, nutzt modernes Webdesign verschiedene Techniken, die zum Teil ineinander verzahnt arbeiten.

Auch wenn der für den Anwender im Browser sichtbare Teil eine Webseite ist, muss man doch eher von einer Web-Anwendung sprechen.

Web-Anwendungen mit HTML, CSS, JavaScript und PHP

Heutige Web-Anwendungen haben immer einen browserseitigen und einen serverseitigen Teil. Eine gebräuchliche Kombination ist HTML, CSS und JavaScript auf Browser-Seite und PHP auf Serverseite.



HTML

In dem hier aufgezeigten Beispiel werden in der HTML-Datei eigentlich nur die anzuzeigenden Elemente im HTML-Format aufgelistet. Dabei wird für die einzelnen Elemente noch keine Eigenschaft wie z.B. Position oder Farbe festgelegt, wie sonst beim klassischen HTML. Stattdessen bekommen die Elemente eine ID (Identifikationsbezeichnung) oder werden einer bestimmten Anzeigeklasse zugeordnet.

```
<span id="footer1" class="blueline">Fußzeile</span>
```

Außerdem gibt es einen Verweis auf die zu verwendende CSS-Datei und das zu nutzende JavaScript.

```
<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 der CSS-Datei werden die visuellen Eigenschaften der in der HTML-Datei gelisteten Elemente zugewiesen.

Das kann individuell für einzelne Elemente erfolgen,

```
#footer1 {
  width:100%;
  height: 20px;
}
```

es können aber auch Zuweisungen für Gruppen von Elementen in Form von Anzeigeklassen festgelegt werden.

```
.blueline {
  color: blue;
}
```

Die Trennung der Style-Eigenschaften von der eigentlichen HTML-Datei ist insbesondere bei größeren Web-Anwendungen sinnvoll, da sich die Style-Definitionen auf mehr als einer Webseite anwenden lassen.

CSS-Dateien haben die Namensendung .css.

JavaScript

JavaScript ist eine Programmiersprache, die im Browser ausgeführt wird. Der JavaScript-Code kann mit in der HTML-Datei integriert sein oder in eine eigene JavaScript-Datei ausgelagert werden.

Der Unterschied zwischen einer Auszeichnungssprache (wie HTML) und einer Programmiersprache liegt im Wesentlichen darin, dass eine Programmiersprache Fallunterschiede berücksichtigt.

Der Code einer Auszeichnungssprache wird stur von oben nach unten ausgeführt. Bei einer Programmiersprache wird nach vorgegebenen Bedingungen entschieden, wann wie was ausgeführt wird.

Mit JavaScript können Elemente, die bereits im Browser angezeigt werden, nachträglich in ihrem Aussehen und ihren Eigenschaften geändert werden. Außerdem können mit JavaScript weitere Daten mit dem Webserver ausgetauscht werden.

JavaScript-Dateien haben die Namensendung .js.

AJAX - Asynchronous JavaScript and XML

AJAX bezeichnet die Technik, mit JavaScript aus einer bereits geladenen Webseite heraus mittels HTTP-Requests mit dem Webserver zu kommunizieren. Das XML in der Bezeichnung AJAX kommt daher, dass viele Web-Anwendungen die Daten mit dem Server im XML-Format austauschen, was aber kein Muss ist.

PHP

Wie JavaScript ist PHP eine Script-Sprache. Allerdings wird der PHP-Programmcode bereits auf dem Server ausgeführt. Der Server erkennt an der Namensendung .php, dass die vom Browser aufgerufene Datei PHP-Anteile enthält, und arbeitet zunächst den PHP-Code ab, der letztlich bestimmt, welche Inhalte an den Browser übertragen werden.

Eine sehr einfache PHP-Datei könnte so aussehen:

```
<!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-Dateien können aufgebaut sein wie eine normale in HTML verfasste Webseite. Die PHP-Anteile sind in entsprechende Tags eingefasst.

- PHP-Beginn `<?php`
- PHP-Ende `?>`

Der so gekennzeichnete PHP-Anteil wird nicht zum Browser geschickt. Im Browser würde für das obige Beispiel folgender Seitenquelltext ankommen:

```
<!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>
```

Durch Ausführen des PHP-Codes wird einfach nur „Willkommen bei WuT.de“ ausgegeben.

PHP-Dateien müssen nicht zwingend HTML-Anteile enthalten. Trotzdem ist der PHP-Quellcode immer von den PHP-Tags eingeschlossen.

Im angeführten Beispiel ergibt der Gebrauch von PHP noch nicht viel Sinn. PHP kann aber deutlich mehr. Eine sehr wichtige Eigenschaft von PHP ist die Datenbankunterstützung. PHP kann direkt auf Datenbanken wie z.B. MySQL zugreifen. Damit bietet es ideale Voraussetzung für die Realisierung von Onlineshops.

Mit PHP lassen sich aber auch normale TCP- oder UDP-Socket-Verbindungen aufbauen und verwalten. Im industriellen Bereich können so auch Daten von Geräten, die kein Web-Interface haben, im Browser darstellt werden.

Neben der hier exemplarisch aufgezeigten Möglichkeit, dynamische Web-Anwendungen zu erstellen, gibt es aber noch weitere Wege, die wir im Folgenden kurz vorstellen wollen.

Serverseitige Programme

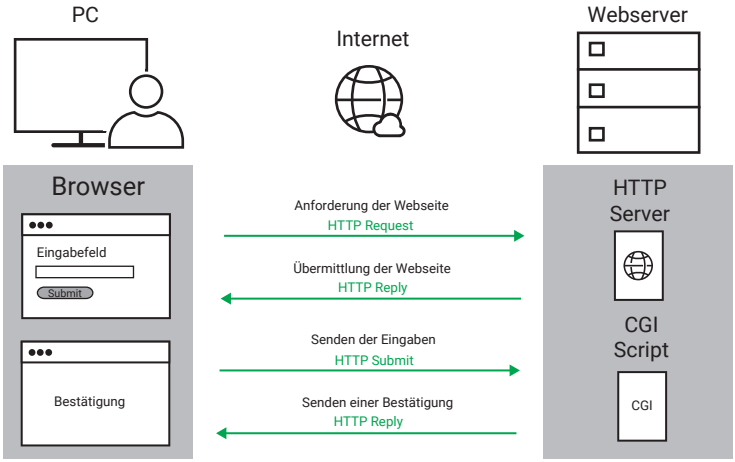
CGI - Common Gateway Interface

Der Einsatz von CGI-Scripten war lange Zeit das meistgenutzte Verfahren, im Browser interaktive Inhalte anzuzeigen bzw. Aktionen auszulösen.

Über CGI können vom Browser aus Programme auf dem Webserver ausgeführt werden.

Über einen Hyperlink, einen Submit-Button oder direkte Eingabe der URL wird das entsprechende Programm aufgerufen und es werden ggf. die nötigen Parameter übergeben.

Ein klassisches Beispiel sind HTML-Formulare, die vom Anwender ausgefüllt werden. Klickt der Anwender den Submit-Button (Abschicken), werden die Eingaben via HTTP mit Hilfe des POST-Kommandos an den Webserver übergeben. Das angegebene CGI-Script wird gestartet und verarbeitet die Eingaben weiter.



Weitere mögliche Anwendungen sind Besucherzähler, Gästebücher, Diskussionsforen, Datenbankzugriffe oder Suchmaschinen.

CGI-Scripte können grundsätzlich in allen gängigen Programmiersprachen erstellt werden. Wichtig ist, dass der Webserver die gewählte Sprache unterstützt.

In der Praxis hat sich die Programmiersprache Perl für die Erstellung von CGI-Scripts durchgesetzt.

PHP

PHP hat CGI heute als meistgenutztes Verfahren für die Darstellung interaktiver Inhalte abgelöst. Eine ausführliche Beschreibung zu PHP gab es ja bereits im vorangegangenen Abschnitt.

ASP - Active Server Pages

ASP ist eine von Microsoft ins Leben gerufene Technik, Webseiten dynamisch anzuzeigen. ASP-basierende Webseiten bestehen wie auch bei der PHP-Technik aus klassischen HTML-Anteilen und Scripten, die bei Aufruf der Webseite serverseitig ausgeführt werden. Ein Interpreter auf dem WWW-Server generiert, gesteuert durch diese Scripte, Webseiten in normalem HTML-Format.

Als Scriptsprachen werden zumeist VBScript (Visual Basic Syntax) oder JScript (Java Syntax) verwendet. Ein Vorteil dieser Technik besteht darin, auf dem Server installierte DLLs und AktivX-Komponenten nutzen zu können. Dynamic Link Libraries und AktiveX-Komponenten sind fertige, ausgelagerte Programmfunktionen, die dem

Programmierer Arbeit abnehmen, da entsprechende, oft komplexe Funktionalitäten nicht selbst programmiert werden müssen.

Der Nachteil von ASP liegt in den Server-Betriebssystemvoraussetzungen. Im Ursprung gab es ASP-Unterstützung nur auf Microsoft Serversystemen. Von Drittherstellern gibt es seit einiger Zeit aber auch ASP-Varianten für Linux-Server.

ASP-basierende Webseiten erkennt man an der Endung „.asp“ im Dateinamen.

Das klassische ASP wurde von Microsoft inzwischen durch ASP.NET abgelöst.

Browserseitige Programme

JavaScript

Eine Beschreibung zu JavaScript gab es ja bereits im vorangegangenen Abschnitt.

Zur Vertiefung hier noch mal ein kurzes Beispiel:

Öffentliche Webauftritte werden über Domainnamen repräsentiert. Dabei kann ein und dieselbe Webseite über mehrere Domainnamen erreichbar sein, zum Beispiel über eine „de“-Domain und eine „com“-Domain. Der folgende Code wertet aus, ob eine Webseite über die „com“-Domain oder die „de“-Domain aufgerufen wurde und stellt sich entsprechend englisch oder deutsch dar.

```
<!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

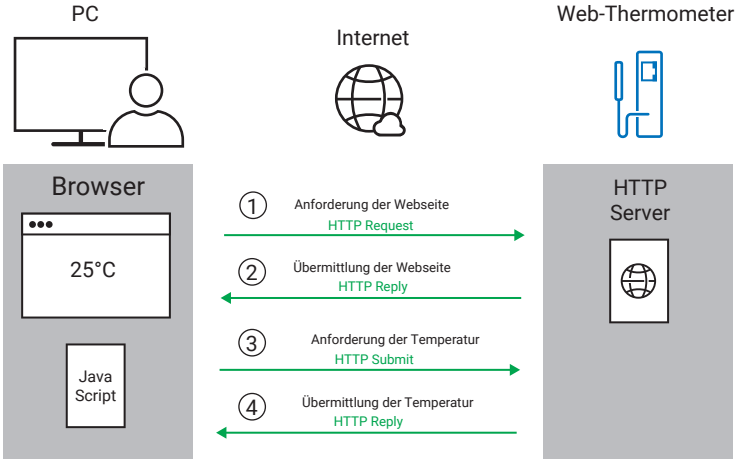
Auch AJAX wurde ja bereits kurz vorgestellt. Da AJAX die zur Zeit populärste Technik ist, von einer bereits geladenen Webseite Daten mit dem Webserver auszutauschen, sollen hier noch ein paar zusätzliche Informationen gegeben werden.

Das Kernstück von AJAX ist die in aktuellen Versionen von JavaScript angebotene HTTP-Request-Methode. Hiermit kann JavaScript auch nach dem Laden und Anzeigen einer Webseite Daten vom Webserver anfordern. Daten können, wenn es der Server unterstützt, im XML-Format angefordert werden. Alternativ wird aber auch die Übertragung von Textformaten unterstützt.

JavaScript übernimmt sowohl das nachträgliche Holen der Daten als auch die Aktualisierung der Browser-Anzeige.

Allerdings kann JavaScript keine dauerhafte Verbindung zum Webserver aufrechterhalten. Es kann aber durch zyklisches Nachladen von Prozessdaten eine selbstaktualisierende Prozessvisualisierung im Browser realisiert werden.

Als kurzes Beispiel hier die Anzeige der Temperatur, die von einem W&T Web-Thermometer erfasst wird:



1. Im Browser wird als URL die vom Anwender hinterlegte Webseite auf dem Web-Thermometer eingegeben und aufgerufen. Der Browser sendet einen entsprechenden HTTP-Request.
2. Das Web-Thermometer sendet die Webseite inklusive des JavaScript-Anteils an den Browser.
3. Das JavaScript wird ausgeführt und fordert per HTTP-Request beim Web-Thermometer die aktuelle Temperatur an.
4. Das Web-Thermometer sendet die aktuelle Temperatur an den Browser, wo das JavaScript die empfangenen Daten auswertet und die Anzeige entsprechend aktualisiert.

Die Schritte 3. und 4. werden solange zyklisch wiederholt, wie die Webseite im Browser geöffnet bleibt.

Aus Sicherheitsgründen erlaubt diese Technik das Nachladen von Daten nur von dem Server, von dem auch die ursprüngliche Webseite mit dem JavaScript geladen wurde. So soll verhindert werden, dass der Browser eines Anwenders dazu genutzt wird, sich unerkannt Zugriff auf die Web-Präsenz Dritter zu verschaffen.

Bei neueren Webservern besteht die Möglichkeit, solche Cross Origin Requests von bestimmten Adressen zuzulassen.

Java-Applets
Noch vor einigen Jahren waren Java-Applets das Mittel der Wahl, wenn es darum ging, dynamische Webseiten zu schaffen. Heute sind Java-Applets von der AJAX-Technik fast vollständig verdrängt worden.

Java-Applets sind kompilierte Programme und benötigen im Browser bestimmte Plugins, also Zusatzfunktionen, die von Hause aus in den aktuellen Browsern nicht vorhanden sind.

Kompilierte Programme bedeutet, die Programmdateien kommen als binäre Daten, sind also anders als z.B. bei JavaScript nicht lesbar.

Binäre Dateien und Plugins sind ein Sicherheitsrisiko, da man nicht sehen kann, welche Funktionen tatsächlich integriert sind.

In vielen Firmennetzen werden Java-Applets deshalb inzwischen von Browsern und Firewalls geblockt.

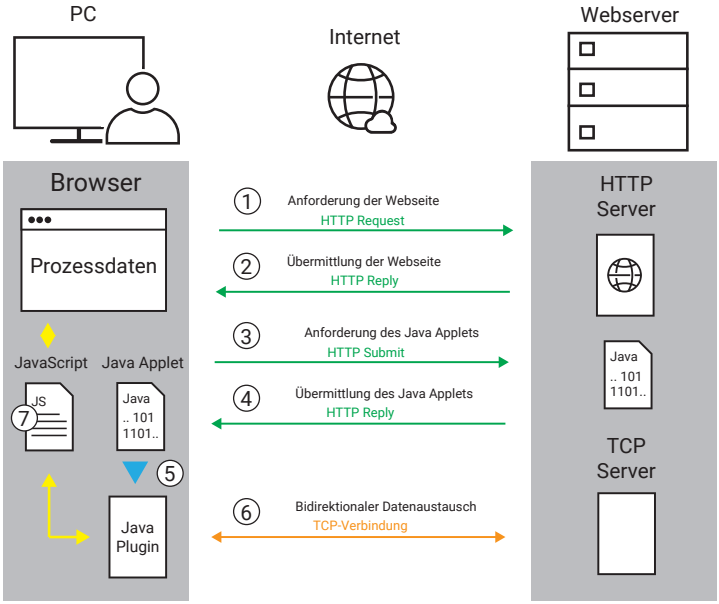
Trotzdem gibt es Fälle, in denen Java-Applets sinnvoll genutzt werden können, wenn sie aus sicherer Quelle kommen.

Ein Nachteil der AJAX-Technik hingegen besteht darin, dass keine permanente Verbindung mit dem Server besteht. Der Server kann von sich aus keine Informationen an den Browser senden und die Kommunikation kann auch nur über HTTP erfolgen.

Mit einem Java-Applet wäre das aber möglich. Der Verweis auf das zu verwendende Java-Applet wird mit in den `<body>`-Bereich der Webseite eingebunden: Dazu werden entsprechende `<applet>`-Tags verwendet. Zusätzlich können noch Parameter für das Applet bestimmt werden.

```
<applet archive="A.jar" code="A.class">
  <param name="getalldata" value="1">
</applet>
```

Hier ein Beispiel, bei dem Prozessdaten über eine normale TCP-Verbindung mit dem Browser ausgetauscht werden sollen:



1. Im Browser wird die URL der Webseite eingegeben und aufgerufen. Der Browser sendet einen entsprechenden HTTP-Request.
2. Der Server sendet die Webseite inklusive eines JavaScript-Anteils an den Browser.
3. Der Browser findet den Verweis auf das Java-Applet und sendet einen zweiten HTTP-Request zum Server.
4. Der Server sendet das Java-Applet zum Browser.
5. Das Java-Applet wird in die Java-Engine geladen und dort gestartet. Die Java-Engine arbeitet als Plugin im Browser.
6. Die Java-Engine baut nun die TCP-Verbindung zum Server auf.
7. Das in die Webseite eingebettete JavaScript tauscht bei Bedarf Sende- und Empfangsdaten der TCP-Verbindung mit der Java-Engine aus. Die empfangenen Daten wertet das JavaScript aus und bringt sie im Browser zur Anzeige.

Neben der Möglichkeit, TCP und UDP als Kommunikationsweg zu nutzen, können Java-Applets auch visuelle Elemente (Anzeigen, Diagramme, Kennlinien, ...) enthalten, die in einer Webseite angezeigt werden sollen.

Schon das einfache Beispiel zeigt, dass die Entwicklung und Einbindung von Webseiten, die ein Java-Applet nutzen, sehr komplex ist. Verbunden mit den bereits genannten Sicherheitsrisiken sollte von Fall zu Fall abgewogen werden, ob Java Applets für die gewünschte Applikation der richtige Weg sind.

Responsives Webdesign

Der Anspruch an Webseiten und die Technik von Webdesign haben sich über die letzten Jahre sehr gewandelt.

In den Anfängen des Internets in HTML erstellte Webseiten kamen eher klotzig daher und waren meist für PC-Monitore mit geringer Auflösung ausgelegt.

Mit JavaScript kam eine gewisse Dynamik dazu und die Inhalte wurden in ihrer Darstellung filigraner, waren aber immer noch für die Ansicht auf PC-Monitoren ausgelegt.

Heute nutzen die meisten Anwender neben dem PC auch Smartphone und Tablet, um Webseiten aufzurufen. Größe und Auflösung der Displays variieren erheblich. Daraus ergeben sich ganz neue Herausforderungen an das Webdesign.

Zwar skalieren die auf dem Smartphone verwendeten Browser die Webseiten automatisch auf die Größe des Displays. Die Inhalte werden dadurch aber häufig so kein, dass sie nicht mehr lesbar sind. Zoomt man die Anzeige größer, sieht man nicht mehr den gesamten Inhalt und die Webseite wird unübersichtlich.

Verschiedene Webseiten für unterschiedliche Displaygrößen

Zunächst haben die Webdesigner darauf mit verschiedenen Webseiten gleichen Inhalts für die verschiedenen Anzeigegeräte reagiert. Per Javascript wurde ermittelt, welches Endgerät die Webseite abruft und dann die passende Webseite an den Browser gesendet.

Das macht die Pflege von Webseiten aber sehr aufwändig, da jeweils zwei bis drei Webseiten angepasst werden müssen, wenn sich der Inhalt ändert.

Responsive Webseiten

Eine deutlich elegantere Methode, den unterschiedlichen Displaygrößen gerecht zu werden, ist reponsives Webdesign.

Wie bereits beschrieben, lassen sich Inhalt und Design einer Webseite durch Cascading-Stylesheet-Dateien trennen. Wie bei normalen Webseiten wird eine HTML- oder PHP-Datei erstellt, in der die Inhalte definiert sind.

Hinzu kommt ein Eintrag im Kopf der Webseite, mit dem festgelegt wird, dass die gesamte Displaybreite mit einer 1:1 Skalierung genutzt wird.

```
<head>
    ........
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    ........
</head>
```

Passend dazu gibt es eine CSS-Datei. Aktuelle Browser akzeptieren innerhalb der CSS-Datei Bereiche mit Stylevorgaben für verschiedene Displaygrößen.

Die Syntax dazu beginnt immer mit `@media` gefolgt von weiteren Parametern. Hier ein Beispiel für eine Displaybreite zwischen 480 Pixeln und 632 Pixeln

```
@media only screen and (min-width: 480px) and (max-width: 632px) {
  .........
}
```

Die entsprechenden Style-Informationen folgen dann eingefasst in geschweifte Klammern.

So können individuell Größe und Position von Anzeigeelementen, Schriftgröße und vieles mehr für die entsprechende Displaygröße vorgegeben werden.

Die Anzeige im Broswer passt sich so automatisch der Bildschirm- bzw. der Größe des Browserfensters an.

Ein weiterer Vorteil dieser Technik ist, dass mit

```
@media only print ....
```

festgelegt werden kann, wie der Ausdruck einer Webseite aussehen soll.

Auch zum Thema Responsives Webdesign gibt es zahlreiche Tutorials im Internet.

Netzwerk-ABC

10Base2 – 10Mbit/s BASEband 200 (185)m/Segment
In den 80er und 90er Jahren wurde 10Base2 als Ethernet-Topologie auf koaxialer Basis zur Vernetzung von PCs und anderen Netzwerkkomponenten eingesetzt.

Die maximale Übertragungsrate war 10Mbit/s.

Weitere Bezeichnungen für 10Base2 waren Cheapernet oder Thin-Ethernet.

Es wurde Koax-Kabel mit 50 Ohm Impedanz in einer dünnen und flexiblen Ausführung verwendet, um die einzelnen Stationen busförmig miteinander zu verbinden. Anfang und Ende eines Segments mussten mit Abschlusswiderständen von 50 Ohm abgeschlossen werden.

Die Schwachstelle der physikalischen Bus-Topologien von Ethernet lag in der Tatsache, dass eine Unterbrechung des Kabels – z.B. durch Abziehen eines Steckverbinders – den Stillstand des gesamten Netzsegmentes zur Folge hatte.

10Base5 – 10Mbit/s BASEband 500m/Segment
10Base5 ist die ursprüngliche Ethernet-Spezifikation. Die Verkabelung beruht auf einem koaxialen Buskabel mit 50 Ohm Impedanz und einer max. zulässigen Länge von 500m (Yellow Cable).

Die Netzwerkteilnehmer wurden über externe Transceiver angeschlossen, die über Vampir-Krallen die Signale direkt vom Buskabel abgriffen, ohne dieses zu unterbrechen. Die Endgeräte wurden über ein zusätzliches Kabel mit dem Transceiver verbunden.

Durch die Verwendung von relativ hochwertigem Kabel ohne jegliche Unterbrechungen durch Steckverbinder ergeben sich die Vorteile der großen Segmentlänge und der hohen Anzahl möglicher Anbindungen pro Segment (max. 100).

Die Dicke und Unflexibilität des Yellow Cable sowie die durch externe Transceiver zusätzlich entstehenden Kosten sind die Hauptnachteile von 10Base5 und haben wohl entscheidend zur Einführung von 10Base2 beigetragen.

10BaseT – 10Mbit/s BASEband Twisted Pair

Mit der Definition von 10BaseT wird die physikalische Topologie von der logischen getrennt. Die Verkabelung ist, ausgehend von einem Hub als zentrale aktive Komponente, sternförmig ausgeführt. Es wird ein mindestens zweipaariges Kabel der Kategorie 3 mit 100 Ohm Impedanz verwendet, in dem die Daten getrennt nach Sende- und Empfangsrichtung übertragen werden. Als Steckverbinder werden 8-polige RJ45-Typen eingesetzt, in denen die Paare auf den Pins 1/2 und 3/6 aufgelegt sind. Die max. Länge eines Segments (= Verbindung vom Hub zum Endgerät) ist auf 100m begrenzt. Ihren Ursprung hat die 10BaseT-Topologie in den USA, weil sie ermöglichte, die dort üblichen Telefonverdrahtungen auch für den Netzwerkbetrieb zu nutzen. Für Deutschland entfiel dieser Vorteil, da hier für die Telefonie Stern-4er-Kabel verlegt wurden, die den Anforderungen der Kategorie 3 nicht entsprachen.

Kabelunterbrechungen oder abgezogene Stecker, die bei allen physikalischen Busstrukturen einen Stillstand des gesamten Segmentes bedeuten, betreffen bei 10BaseT lediglich einen Arbeitsplatz. (*Siehe. a.* 🗎 *13)*

100BaseFX – 100Mbit/s BASEband Fiber Exchange

Ethernet-Standard für die sternförmige Glasfaserverkabelung mit einer Übertragungsgeschwindigkeit von 100Mbit/s. Zur Übertragung des Ethernet-Signals werden Lichtimpulse mit einer Wellenlänge von 1300nm in eine 50µm oder 62,5µm Multimodefaser eingespeist. Die maximale Segmentlänge beträgt 2km. (*Siehe. a. LWL)*

100BaseT4 – 100Mbit/s BASEband Twisted 4 Pairs

100BaseT4 spezifiziert eine Ethernet-Übertragung mit 100Mbit/s. Wie bei 10BaseT handelt es sich um eine physikalische Sternstruktur mit einem Hub als Zentrum. Es wird ebenfalls ein Kabel der Kategorie 3 mit 100 Ohm Impedanz, RJ45 Steckverbindern und einer max. Länge von 100m eingesetzt. Die zehnfache Übertragungsgeschwindigkeit von 100Mbit/s bei gleichzeitiger Einhaltung der Kategorie-3-Bandbreite von 25MHz wird u.a. auch durch die Verwendung aller vier Aderpaare erzielt. Für jede Datenrichtung werden bei 100BaseT4 immer 3 Paare gleichzeitig verwendet.

100BaseTX – 100Mbit/s BASEband Twisted 2 Pairs

100BaseTX spezifiziert die 100Mbit/s-Übertragung auf 2 Aderpaaren über eine mit Komponenten der Kategorie 5 realisierte Verkabelung. Kabel, RJ45-Wanddosen, Patchpanel usw. müssen gemäß dieser Kategorie für eine Übertragungsfrequenz von mindestens 100MHz ausgelegt sein.

1000BaseT
1000BaseT wird auch als Gigabit-Ethernet bezeichnet. Über Kabel und Komponenten, die mindestens der Kategorie 5 entsprechen, können über eine maximale Distanz von 100m 1000MBits/s übertragen werden. (*Siehe. a. 14)*

1000BaseBX, 1000BaseLX, 1000BaseSX, 1000Base ZX/EZX
Neben dem drahtgebundenen 1000BaseT gibt es diverse Gigabit-Ethernet-Standards, die Glasfaser als Übertragungsmedium nutzen. *(Siehe a. LWL)*

10GBaseT
Mit Übertragungsraten bis zu 10GBits/s wird 10GBaseT als Backbone, also Hintergrundverkabelung zwischen Swichtes verwendet. Es werden Kabel von mindestens Kategorie 6 benötigt. (*Siehe. a. 15)*

10GBaseER, 10GBaseEW, 10GBaseLX4, 10GBase LW, 10GBaseSW
Übertragungsstandards für 10GBase über Glasfaser. *(Siehe a. LWL)*

ABAP - Advanced Business Application Programming
ABAP ist eine Programmiersprache, die von SAP entwickelt wurde, um die SAP-Softwareumgebung individuell zu programmieren.

Abschlusswiderstand
Bei koaxialen Netzwerktopologien wie 10Base5 oder 10Base2 muss jeder Netzwerkstrang am Anfang und am Ende mit einem Abschlusswiderstand (Terminator) abgeschlossen werden. Der Wert des Abschlusswiderstandes muss der Kabelimpedanz entsprechen; bei 10Base5 oder 10Base2 sind dies 50 Ohm.

Administrator
Systemverwalter, der im lokalen Netzwerk uneingeschränkte Zugriffsrechte hat und für die Verwaltung und Betreuung des Netzwerks zuständig ist. Der Administrator vergibt unter anderem die IP-Adressen in seinem Netzwerk und muss die Einmaligkeit jeder IP-Adresse gewährleisten.

ADSL
Asynchroner DSL-Anschluss mit unterschiedlichen Geschwindigkeiten für Up- und Download. (*Siehe. a. 103)*

AES
Advanced Encryption Standard ist ein symmetrischer Verschlüsselungsalgorithmus, der z.B. bei der Übertragung von Webseiten über HTTPS verwendet wird.
(*Siehe. a. 155)*

AJAX - Asynchronous JavaScript and XML

AJAX ist eine JavaScript-Programmiertechnik, die es erlaubt, Inhalte einer Webseite durch Nachladen von Daten dynamisch zu aktualisieren. (*Siehe. a.* 📄 *215ff)*

ARP – Address Resolution Protocol

Über ARP wird die zu einer IP-Adresse gehörende Ethernet-Adresse eines Netzwerkteilnehmers ermittelt. Die ermittelten Zuordnungen werden auf jedem einzelnen Rechner in der ARP-Tabelle verwaltet. In Windows-Betriebssystemen kann man auf die ARP-Tabelle mit Hilfe des ARP-Befehls Einfluss nehmen. Eigenschaften und Parameter des ARP Kommandos in der DOS-Box:

- ARP -A listet die Einträge der ARP-Tabelle auf
- ARP -S <IP-Adresse> <Ethernet-Adresse> fügt der ARP-Tabelle einen statischen Eintrag hinzu
- ARP -D <IP-Adresse> löscht einen Eintrag aus der ARP-Tabelle

ARP ist im Internet-Standard RFC-826 definiert. (*Siehe. a.* 📄 *26)*

ASCII-Kodierung

Mit dem American Standard Code for Information Interchange wurde bereits 1963 festgelegt, mit welchem 7-Bit-Wert welches Zeichen bei der Datenübertragung kodiert wird.

0x0 0000	0x1 0001	0x2 0010	0x3 0011	0x4 0100	0x5 0101	0x6 0110	0x7 0111	hex. dual
NUL 0	DEL 16	Space 32	0 48	@ 64	P 80	` 96	p 112	0x0 0000
SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113	0x1 0001
STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114	0x2 0010
ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115	0x3 0011
EOT 4	DC4 20	$ 36	4 52	D 68	T 84	d 100	t 116	0x4 0100
ENQ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117	0x5 0101
ACK 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118	0x6 0110
BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119	0x7 0111
BS 8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120	0x8 1000
TAB 9	EM 25	) 41	9 57	I 73	Y 89	i 105	y 121	0x9 1001
LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122	0xA 1010
VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123	0xB 1011
FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	| 124	0xC 1100
CR 13	GS 29	- 45	= 61	M 77	] 93	m 109	} 125	0xD 1101
SO 14	RS 30	. 46	> 62	N 78	^ 94	n 110	~ 126	0xE 1110
SI 15	US 31	/ 47	? 63	O 79	_ 95	o 111	DEL 127	0xF 1111

Neben den darstellbaren Zeichen enthält die Kodierung für die Werte 0 bis 32 und den Wert 127 funktionale Codes:

0 NUL NULL-Zeichen. Wird verwendet um Abschnitte in Datenströmen zu Kennzeichnen
1 SOH Start Of Heading - Erstes Zeichen des Nachrichtenkopfes
2 STX Start Of Text - Erstes Zeichen einer Nachricht
3 ETX End Of Text - Ende einer Nachricht
4 EOT End Of Transmission - Ende der Datenübertragung
5 ENQ Enquiry - Anforderung einer Rückmeldung
6 ACK Acknowledgment - Datenempfangsquittierung

7 BEL Klingel/Glocke
8 BS Backspace
9 TAB Tabulator horizontal
10 LF Line Feed - Zeilenvorschub
11 VT Tabulator vertikal
12 FF Form Feed - Seitenvorschub
13 CR Carriage Return - setzt den Cursor auf die erste Position in der Zeile
14 SO Shift out - Beendet mit Shift in begonnene Sonderbehandlung
15 SI Shift in - nachfolgende Zeichen werden besonders behandelt
16 DLE Data Link Escape
17 DC1 Device Control 1 - ruft vorgegebene Gerätefunktion auf
18 DC2 Device Control 2 - ruft vorgegebene Gerätefunktion auf
19 DC3 Device Control 3 - ruft vorgegebene Gerätefunktion auf
20 DC4 Device Control 4 - ruft vorgegebene Gerätefunktion auf
21 NAK Negative Acknowlegdement - Datenempfang nicht korrekt
22 SYN Synchronous Idle
23 ETB End Of Transmission Block - Ende eines Übertragungsblocks
24 CAN Cancel
25 EM End Of Medium - Keine weitere Datenverarbeitung möglich
26 SUB Substitute
27 ESC Escape - Auf Escape folgende Zeichen haben eine spezielle Bedeutung
28 FS File Separator - Trennzeichen
29 GS Group Separator- Trennzeichen
30 RS Record Separator- Trennzeichen
31 US Unit Separator- Trennzeichen
127 DEL Delete - stammt aus der Zeit, als Daten auf Lochkarten gespeichert wurden. Die sieben Bits waren mit 1 = Loch bzw. 0 = nicht Loch kodiert. Mit DEL wurden alle 7 Löcher gestanzt. So konnte jedes Zeichen überschrieben und damit als ungültig gekennzeichnet werden

ASN.1

Format für den Aufbau von SNMP-MIB-Dateien. (*Siehe. a.* *87)*

AUI – Attachment Unit Interface

Schnittstelle zur Anbindung eines externen Ethernet-Transceivers.

Getrennt nach Sende-, Empfangs- und Kollisions-Information werden die Daten vom Transceiver auf einem 15-poligen D-SUB-Steckverbinder zur Verfügung gestellt. Der Anschluss des Endgerätes erfolgt über ein 8-adriges TP-Kabel von max. 50m Länge. Während die AUI-Schnittstelle in der Vergangenheit hauptsächlich zur Ankopplung von Endgeräten an 10Base5-Transceiver (Yellow-Cable) genutzt wurde, verwendet

man sie heute eher zur Anbindung an LWL-Transceiver (Glasfaser) o.ä.

Backbone

Als Backbone wird die Hintergrundverkabelung zwischen Standorten bzw. Switches bezeichnet. Oft wird für die Backbone-Verbindungen ein schnelleres Übertragungsverfahren gewählt als für die Anschlüsse innerhalb des lokalen Netzwerks.

Base64

Kodierungsverfahren, um binäre Daten wie z.B. Bilder mit 7Bit-Technik zu übertragen. (*Siehe. a.* 📄 *121)*

Bit

Das Bit ist die kleinste Speichereinheit in der Computertechnik und kann die beiden Zustände 1 oder 0 annehmen. (*Siehe. a.* 📄 *9, 256)*

Bluetooth

Funkstandard, um Endgeräte über kurze Distanzen miteinander zu verbinden.

Binärdaten

Binäre Daten sind Daten, bei denen jedes Byte Werte zwischen 0 und 255 haben darf. (*Siehe. a.* 📄 *116)*

BNC – Bayonet Neill Concelmann

Bei der BNC-Steckverbindung handelt es sich um einen Bajonettverschluss zum Verbinden zweier Koaxialkabel. BNC-Steckverbindungen werden in 10Base2-Netzwerken zur mechanischen Verbindung der RG-58-Kabel (Cheapernet) verwendet.

BootP – Boot Protocol

Dieses ältere Protokoll zum Booten von PCs ohne Festplatte über das Netzwerk ist der Vorläufer von DHCP. Auch moderne DHCP-Server unterstützen immer noch BootP-Anfragen. Heute wird BootP in erster Linie eingesetzt, um Embedded Systemen eine IP-Adresse zuzuteilen. Dazu muss auf dem DHCP-Server ein reservierter Eintrag hinterlegt werden, in dem der MAC-Adresse des Embedded Systems eine feste IP-Adresse zugeordnet ist.

Bridge

Bridges verbinden Teilnetze miteinander und entscheiden anhand der Ethernet-Adresse, welche Pakete die Bridge passieren dürfen und welche nicht. Die dazu notwendigen Informationen entnimmt die Bridge Tabellen, die je nach Modell vom Administrator eingegeben werden müssen oder von der Bridge dynamisch selbst erstellt werden. *(Siehe a. Router)*

Broadcast

Als Broadcast bezeichnet man einen Rundruf an alle Netzteilnehmer. Eine typische Broadcast-Anwendung ist der ARP-Request (siehe ARP). Auch andere Protokolle – etwa RIP oder DHCP – nutzen Broadcast-Meldungen.

Broadcast-Meldungen werden nicht über Router oder Bridges weitergegeben.

Broker

Als Broker bezeichnet man einen Server innerhalb einer MQTT-Anwendung, der Daten per Publish/Subscribe-Verfahren weitervermittelt. (*Siehe. a.* 132)

Browser

Client-Programm mit grafischer Benutzeroberfläche, das dem Anwender die Möglichkeit gibt, Webseiten anzuzeigen und andere Dienste im Internet zu nutzen.
(*Siehe. a.* 208)

Bussystem

Bei einem Bus-System teilen sich mehrere Endgeräte eine einzige Datenleitung (Busleitung). Da zu einer gegebenen Zeit jeweils nur ein Endgerät die Datenleitung benutzen darf, erfordern Bus-Systeme immer ein Protokoll zur Regelung der Zugriffsrechte. Klassische Bus-Systeme sind die Ethernet-Topologien 10Base2 und 10Base5.

Byte

Ein Byte besteht aus 8 Bits und ist die kleinste Datenmenge, die Computer verarbeiten können. Aus der Breite von 8 Bits ergibt sich, dass mit einem Byte Zahlenwerte zwischen 0 und 255 gespeichert bzw. übertragen werden können. Mehr dazu im Kapitel Zahlensysteme. (*Siehe. a.* 9)

Cache

Als Cache bezeichnet man einen Zwischenspeicher, wie er z.B. im Browser zum Einsatz kommt, um Webseiten und andere Inhalte vorübergehend festzuhalten. Wird ein Inhalt in kurzen Abständen mehrfach beim Webserver abgerufen, fordert der Browser die benötigten Daten nicht erneut beim Server an, sondern nimmt die bereits geladenen aus dem Cache. (*Siehe. a.* 100)

Cheapernet

Andere Bezeichnung für Ethernet auf der Basis von 10Base2.

Checksumme

Über den Inhalt von übertragenen oder gespeicherten Daten kann nach einem vorgegebenen Algorithmus eine Checksumme gebildet werden. Vor der Weiterverarbei-

tung der Daten kann der gleiche Algorithmus erneut angewendet werden, um zu prüfen, ob der Inhalt unverändert ist. (*Siehe. a. 149)*

Cipher Suites

Als Cipher Suite wird eine Kombination aus Verfahren zur Authentifizierung, Integritätsprüfung und Verschlüsselung von Daten bezeichnet. (*Siehe. a. 174)*

Client

Computer oder Anwendungen, die Dienste von sogenannten Servern in Anspruch nehmen. Serverdienste können zum Beispiel die Bereitstellung einer COM- oder Drucker-Schnittstelle im Netzwerk, aber auch Telnet und FTP sein. (*Siehe. a. 28)*

Client/Server-Architektur

System der „verteilten Intelligenz", bei dem der Client eine Verbindung zu einem Server aufbaut, um vom Server angebotene Dienste in Anspruch zu nehmen. Manche Server-Anwendungen können mehrere Clients gleichzeitig bedienen. (*Siehe. a. 28)*

Com-Server

Endgerät in TCP/IP-Ethernet-Netzwerken, das Schnittstellen für serielle Geräte über das Netzwerk zur Verfügung stellt. *(Siehe a. https://www.wut.de/58665)*

Community String

Der Community String ist eine Art Passwort, das bei jeder SNMP-Abfrage mitgesendet wird. (*Siehe. a. 91)*

Cookies

Nutzerinformationen, wie z.B. Kundennummer o.ä., die der Browser derart zwischenspeichert, dass sie nach dem nächsten Start beim Besuch der gleichen Webseite noch erhalten sind.

Cross-Link-Kabel

Netzwerkkabel, bei dem die Kabeladern für Senden und Empfangen gekreuzt sind. Mit einem Cross-Link-Kabel können Netzwerkendgeräte (die kein Autocrossing unterstützen) ohne zusätzlichen Switch direkt verbunden werden.

DHCP – Dynamic Host Configuration Protocol

Dynamische Zuteilung von IP-Adressen aus einem Adressenpool.

DHCP wird benutzt, um PCs in einem TCP/IP-Netz automatisch – also ohne manuellen Eingriff – zentral und somit einheitlich zu konfigurieren. Der Systemadministrator bestimmt, wie die IP-Adressen zu vergeben sind, und legt fest, über welchen Zeit-

raum sie vergeben werden. DHCP ist in den Internet-Standards RFC 2131 (03/97) und RFC 2241 (11/97) definiert. (*Siehe. a.* 59)

DDNS – Dynamic Domain Name Service
DNS-Dienst, der auch die Namensauflösung für solche Netzteilnehmer unterstützt, die ihre IP-Adresse dynamisch über DHCP beziehen. (*Siehe. a.* 68)

DNS – Domain Name Service
Netzteilnehmer werden im Internet über numerische IP-Adressen angesprochen. Doch weil man sich Namen eben besser merken kann als Nummern, wurde der DNS eingeführt.

DNS beruht auf einem hierarchisch aufgebauten System: Jede Namensadresse wird über eine Top-Level-Domain („de", „com", „net" usw.) und innerhalb dieser über eine Sub-Level-Domain identifiziert. Jede Sub-Level-Domain kann (muss aber nicht) nochmals untergeordnete Domains enthalten. Die einzelnen Teile dieser Namenshierarchie sind durch Punkte voneinander getrennt.

Wird vom Anwender zur Adressierung ein Domain-Name angegeben, erfragt der TCP/IP-Stack beim nächsten DNS-Server die zugehörige IP-Adresse.

Netzwerkressourcen sollten sinnvollerweise einen Domainnamen erhalten, der im Kontext zu der angebotenen Dienstleistung oder dem Firmennamen des Anbieters steht. So lässt sich z.B. wut.de in die Top-Level-Domain de (= Deutschland) und die Sub-Level-Domain wut (= Wiesemann & Theis GmbH) auflösen. (*Siehe. a.* 64)

DNS-Server
DNS-Server stellen im Internet die Dienstleistung zur Verfügung, einen Domain-Namen in eine IP-Adresse aufzulösen.

DOS Disk-Operation-System
Frühes Betriebssystem von Microsoft auf Kommandozeilenbasis.

DynDNS
Bei den meisten Internetzugängen bekommt das angeschlossene Endgerät zum Zeitpunkt der Einwahl eine IP-Adresse aus dem Adresspool des Internet-Providers. Da diese temporäre IP-Adresse nach außen nicht bekannt ist, sind solche Endgeräte normalerweise vom Internet aus nicht adressierbar. Über DynDNS kann einem solchen Internetteilnehmer ein Name gegeben werden. DynDNS aktualisiert die Zuordnung zwischen Namen und temporärer IP-Adresse, sobald der Teilnehmer online geht, so dass eine Erreichbarkeit über den Namen möglich wird. (*Siehe. a.* 69)

EDGE - Enhanced Data Rates for GSM Evolution

EDGE ist eine Weiterentwicklung der GSM-Technik und basiert auf effizienteren Datenmodulations- bzw. Kompressionsverfahren. GPRS wird mit EDGE zu E-GPRS (Enhanced GPRS) und erlaubt Datenraten bis zu 220kbit/s (Download) bzw. 110kbit/s (Upload).

Da EDGE nur eine Erweiterung von GSM ist, können im gleichen Netz Endgeräte beider Techniken betrieben werden.

E-Mail

Elektronische Post über Internet und Intranet. (*Siehe. a.* 📄 *103)*

E-Mail-Adresse

Eine E-Mail-Adresse wird benötigt, um einem Anwender elektronische Post senden zu können und setzt sich immer aus dem Mailbox-Namen des Anwenders und der Ziel-Domain, getrennt durch das @-Zeichen zusammen. Ein Beispiel: info@wut.de bezeichnet das Info-Postfach auf dem Mailserver von W&T. (*Siehe. a.* 📄 *104)*

Embedded System

Als Embedded System bezeichnet man eine mikroprozessorgesteuerte Baugruppe, die als eingebetteter Teil eines Gerätes oder einer Maschine im Hintergrund Daten verarbeitet und ggf. Prozesse steuert.

ERP-System

Enterprise Resource Planning System - darunter versteht man eine Software-Lösung, die Unternehmen hilft, Kapital, Betriebsmittel und Personal möglichst effizient zu nutzen. Bekanntester Anbieter auf diesem Gebiet ist SAP.

Ethernet

Ethernet ist die zur Zeit bei lokalen Netzen am häufigsten angewandte Technologie. (*Siehe. a.* 📄 *12ff)*

Ethernet-Adresse

Die unveränderbare physikalische Adresse einer Netzwerkkomponente im Ethernet.

(*Siehe. a.* 📄 *22)*

Fast-Ethernet

Fast-Ethernet ist quasi ein Upgrade der 10BaseT-Topologie von 10MBits/s auf 100 Mbit/s. *(Siehe hierzu a.100BaseT4 und 100BaseTX)*

Feldbus
Bussystem für industriellen Einsatz *(Siehe Bussysteme)*

Firewall
Unter einer Firewall versteht man eine Netzwerkkomponente, die ähnlich einem Router zwei Netzwerke miteinander verbindet und dabei aber nach vorgegebenen Regeln filtert, welche Inhalte von einem Netzwerk ins andere übertragen werden dürfen. Oft sind Firewalls Bestandteil von Routern, die ein internes Netzwerk (Intranet) an ein öffentliches Netzwerk (z.B. Internet) ankoppeln. (*Siehe. a. 47)*

FTP – File Transfer Protocol
FTP ist ein auf TCP/IP aufsetzendes Protokoll, das es ermöglicht, ganze Dateien zwischen zwei Netzwerkteilnehmern zu übertragen. (*Siehe. a. 79)*

Gateway
Gateways verbinden – wie auch Bridges und Router – verschiedene Netze miteinander. Während Bridges und Router zwar ggf. die physikalische Art des Netzes umsetzen (z.B. Ethernet/ISDN), das eigentliche Protokoll (z.B. TCP/ IP) aber unberührt lassen, bieten Gateways die Möglichkeit, einen Zugang zu protokollfremden Netzen zu schaffen (z.B. TCP/IP auf Profibus). Ein Gateway hat also unter anderem auch die Aufgabe, unterschiedliche Kommunikationsprotokolle zu übersetzen.

Achtung: Bei der Netzwerkkonfiguration in Windows-Betriebssystemen wird auch die Eingabe eines Gateways gefordert. Diese Angabe bezieht sich allerdings auf einen ggf. im Netzwerk vorhandenen Router! (*Siehe. a. 39)*

GPRS - General Packet Radio Service
GPRS ist ein auf GSM aufsetzender Standard zur Datenübertragung.

Obwohl der Fokus bei GSM auf die Übertragung von Sprachdaten gelegt wurde, bietet GSM die Möglichkeit, mittels GPRS Daten zu übertragen. Auch für die Datenübertragung stehen pro Frequenz 8 Kanäle zur Verfügung. Bei der Telefonie wird für die Dauer eines Gespräches eine Verbindung aufgebaut, die in jede Richtung einen Kanal blockiert.

Bei der Datenübertragung mit GPRS wird ein Kanal nur dann blockiert, wenn wirklich Daten gesendet werden, und kann so zeitversetzt von mehreren Teilnehmern genutzt werden. Auch die parallele Nutzung mehrerer Kanäle durch einen Teilnehmer ist zulässig. So können Übertragungsraten von bis zu 54kbits/s erreicht werden, was in etwa der Übertragungsgeschwindigkeit analoger Modems entspricht.
(*Siehe. a. 204ff)*

GSM - Global System for Mobile Communications

GSM ist der ursprüngliche und erste Standard der digitalen Mobilfunktelefonie und technische Grundlage der D- und E-Netze. GSM arbeitet in einem Frequenzbereich zwischen 890MHz und 960MHz. Innerhalb dieses Bereiches gibt es je 124 Kanäle pro Übertragungsrichtung. Innerhalb jeder Einzelfrequenz teilen sich 8 Kanäle die Übertragungszeit. Durch Datenkompressionsverfahren können so 8 Teilnehmer gleichzeitig über dieselbe Sende- bzw. Empfangsfrequenz telefonieren.
(*Siehe. a.* 204ff)

HSPA, HSDPA/HSUPA - High Speed Packet Access

HSPA ist eine Weiterentwicklung von UMTS. Es wird das gleiche Frequenzband und die gleiche Sendetechnik auf Provider-Seite verwendet. Dabei kommt allerdings ein deutlich verbessertes Modulations- und Kodierungsverfahren zum Einsatz. Um ein Maximum an Effizienz zu erreichen, wird beim Download (HSDPA) anders gearbeitet als beim Upload (HSUPA). Der Grund hierfür liegt darin, dass die Sendeeinrichtung auf Provider-Seite deutlich mehr Sendeleistung zur Verfügung hat als das mobile Endgerät des Kunden.

Beim Download mit HSDPA können bis zu 42MBit/s erreicht werden. Beim Upload sind bis zu 5,8MBit/s möglich. (*Siehe. a.* 204ff)

HTML – Hypertext Markup Language

Auszeichnungssprache, die über Schlüsselwörter vorgibt, wie die Inhalte im Browser angezeigt werden, wo Multimedia-Elemente zu finden sind und welche Elemente wie verlinkt sind. *(Siehe. a.* 210ff)

HTTP – Hypertext Transfer Protocol

Das HTTP-Protokoll setzt auf TCP auf und regelt die Anforderung und Übertragung von Web-Inhalten zwischen HTTP-Server und Browser. (*Siehe. a.* 94)

Hyperlink

Verweis auf andere Webseiten oder Inhalte innerhalb einer Webseite. Durch einfaches Anklicken des verlinkten Elements gelangt der Anwender auf die gewünschte Webseite. (*Siehe. a.* 212)

Hub

Ein Hub – oft auch als Sternkoppler bezeichnet – bietet die Möglichkeit, mehrere Netzteilnehmer sternförmig miteinander zu verbinden. Datenpakete, die auf einem Port empfangen werden, werden gleichermaßen auf allen anderen Ports ausgegeben.

Neben Hubs für 10BaseT (10Mbit/s) und 100BaseT (100Mbit/s) gibt es sogenannte Autosensing-Hubs, die automatisch erkennen, ob das angeschlossene Endgerät mit 10 oder 100Mbit/s arbeitet. Über Autosensing-Hubs können problemlos ältere 10BaseT-Geräte in neue 100BaseT-Netzwerke eingebunden werden. (*Siehe. a.* 15)

ICMP – Internet Control Message Protocol

Das ICMP-Protokoll dient der Übertragung von Statusinformationen und Fehlermeldungen zwischen IP-Netzknoten. ICMP bietet außerdem die Möglichkeit einer Echo-Anforderung (siehe auch Ping); auf diese Weise lässt sich feststellen, ob ein Bestimmungsort erreichbar ist. (*Siehe. a.* 73)

Internet

Das Internet ist der weltweit größte Netzverbund, der den angeschlossenen Netzteilnehmern eine nahezu grenzenlose Kommunikationsinfrastruktur zur Verfügung stellt. Durch Einsatz von TCP/IP können die Netzteilnehmer plattformunabhängig im Internet angebotenen Dienste wie E-Mail, FTP, HTTP usw. in Anspruch nehmen.

Intranet

Ein abgeschlossenes Netzwerk (etwa innerhalb eines Unternehmens), in dessen Grenzen die Netzteilnehmer internet-typische Dienste wie E-Mail, FTP, HTTP usw. in Anspruch nehmen können. In aller Regel gibt es von einem Intranet über Router bzw. Firewalls auch Übergänge in das Internet.

IP – Internet Protocol

Protokoll, das die Verbindung von Teilnehmern ermöglicht, die in unterschiedlichen Netzwerken positioniert sind. (*Siehe. a.* 24ff)

IP-Adresse

Die IP-Adresse ist eine 32-Bit-Zahl, die jeden Netzteilnehmer im Internet bzw. Intranet eindeutig identifiziert. Sie besteht aus einem Netzwerkteil (Net-ID) und einem Benutzerteil (Host-ID). (*Siehe. a.* 25)

IPX

Steht für Internet Packet Exchange und wurde von Novell als Netzwerkprotokoll für Novell-Netware entwickelt.

IPsec

IPsec ist ein Protokoll, das lokale Netzwerke gesichert und verschlüsselt über öffentliche Netzwerke wie das Internet verbindet. IPsec wird beim Aufbau von VPNs (Virtual Private Networks) eingesetzt. (*Siehe. a.* 193)

ISDN – Integrated Services Digital Network

ISDN wurde in den 1980er Jahren als neuer Standard in der Fernmeldetechnik eingeführt. Bei ISDN werden Telefon und Telefax, aber auch Bildtelefonie und Datenübermittlung integriert. Über ISDN können also abhängig von den jeweiligen Endgeräten Sprache, Texte, Grafiken und andere Daten übertragen werden.

ISDN stellt über die S0-Schnittstelle eines Basisanschlusses zwei Basiskanäle (B-Kanäle) mit je 64 kbit/s sowie einen Steuerkanal (D-Kanal) mit 16 kbit/s zur Verfügung. Der digitale Teilnehmeranschluss hat zusammengefasst eine maximale Übertragungsgeschwindigkeit von 144 kbit/s (2B+D). In den beiden B-Kanälen können gleichzeitig zwei unterschiedliche Dienste mit einer Bitrate von 64 kbit/s über eine Leitung bedient werden. (*Siehe. a.* 200)

ISDN-Router

ISDN-Router gestatten es, zwei lokale Netzwerke über das ISDN-Netz eines Telefonnetz-Providers miteinander zu verbinden. Dabei übernehmen ISDN-Router neben den normalen Funktionen eines Routers auch das Handling der ISDN-Verbindung.

JSON

Auszeichnungssprache, deren Syntax an JavaScript angelehnt ist. (*Siehe. a.* 119)

L2TP - Layer 2 Tunneling Protocol

Mit dem L2TP-Protokoll können Daten zwischen zwei Netzwerken unverschlüsselt getunnelt werden. (*Siehe. a.* 195)

LAN – Local Area Network

Lokales Netz innerhalb eines begrenzten Gebiets unter Anwendung eines schnellen Übertragungsmediums wie z.B. Ethernet.

LTE - Long Term Evolution

In Verbindung mit LTE wird oft von der vierten Mobilfunkgeneration gesprochen, was nicht ganz richtig ist. LTE ist ein reines IP-basierendes Datennetz. Für die Telefonie nutzen LTE-fähige Endgeräte z.Zt. immer noch das ganz normale GSM/UMTS-Netz.

Mit LTE soll erstmals ein international einheitlicher Mobilfunkstandard geschaffen werden. Deshalb ist LTE, was die benutzten Funkfrequenzen angeht, flexibel. In Deutschland werden zwei Frequenzbereiche genutzt:

- 800MHz
- 2,6GHz

Das 800MHz-Frequenzband wurde ehemals für die Übertragung analoger Fernsehkanäle genutzt und ist mit dem Wegfall dieser Technik freigeworden. Ein großer Vorteil dieses Frequenzbandes ist die große Reichweite von bis zu 30km. Damit können auch ländliche Gebiete gut mit LTE abgedeckt werden. Das 2,6GHz Frequenzband kommt vorrangig in Ballungsgebieten mit räumlich kleineren Funkzellen zum Einsatz. Ein effizienteres Kodierungsverfahren und eine deutlich verbesserte Technik auf Provider-Seite erlauben Übertragungsraten von bis zu 100MBit/s (theoretisch sogar 300MBit/s) beim Download und 50MBit/s (theoretisch sogar 100MBit/s) beim Upload.

Durch die hohen Übertragungsgeschwindigkeiten bietet sich LTE als Alternative zum DSL-Anschluss an. Allerdings teilt sich die gesamte in einer Funkzelle verfügbare Bandbreite, wie bei den anderen Mobilfunktechniken auch, auf die Anzahl der aktiven Nutzer auf. (*Siehe. a.* 📄 *204ff)*

LWL - Lichtwellenleiter

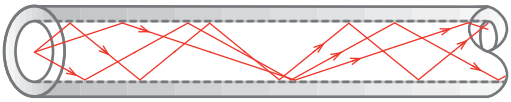
In der Netzwerk- und Nachrichtentechnik werden zunehmend Lichtwellenleiter - kurz LWL - als Kommunikationsmedium eingesetzt. Vor allem in der Netzwerktechnik lassen sich mit LWL deutlich größere Distanzen überbrücken als mit herkömmlicher Kupferverkabelung. Darüber hinaus ist die Datenübertragung über LWL resistent gegen elektrische Einflüsse wie z.B. Blitzschlag und Einkoppelung von Fremd- und Störsignalen.

Elektrische Signale werden in Lichtsignale gewandelt und über LWL-Transmitter in den Lichtwellenleiter eingespeist. Als Übertragungsmedium werden meist Glasfasern eingesetzt.

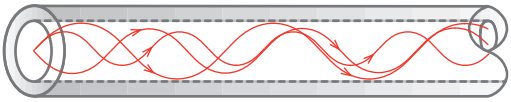
Dabei unterscheidet man zwischen Multimodefasern, Monomodefasern und Kunststofflichtwellenleiter.

Multimodefasern

Multimodefasern haben einen Faserdurchmesser von 62,5µm oder 50 µm. Da sich Licht, wenn möglich, in alle Richtungen ausbreitet, nimmt es innerhalb der Faser verschiedene Signalwege (deshalb Multimode). Durch die unterschiedlichen Reflexionswinkel legt das Licht kürzere und längere Wege zurück, bis es beim Empfänger ankommt.



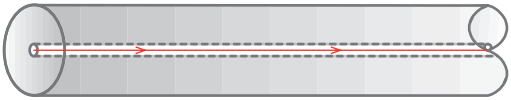
Solche Multimode-LWL werden auch als Stufenindexfasern bezeichnet. Neben den Stufenindexfasern gibt es Gradientenindexfasern. Auch bei diesen Fasern breitet sich das Licht in verschiedene Richtungen aus. Durch eine besondere optische Beschaffenheit werden die Lichtstrahlen aber sanft abgelenkt und nicht wie bei der Stufenindexfaser vom Rand reflektiert.



Gradientenindexfasern haben eine höhere Bandbreite als Stufenindexfasern und erlauben deshalb höhere Signalgeschwindigkeiten.

Mit beiden Multimodefasertypen können, abhängig vom zu übertragenen Signal, Distanzen von bis zu mehreren Kilometern überbrückt werden (bei 100BaseFX z.B. max. 2km).

Monomodefasern
Monomodefasern - oft auch als Singlemodefasern bezeichnet - haben einen Faserdurchmesser von 3-9µm. Bedingt durch den geringen Faserdurchmesser kann sich das Licht nur auf einem Signalweg ausbreiten (deshalb Monomode).



Monomodefasern erlauben je nach zu übertragendem Signal Distanzen von bis zu 50km.

Durch den geringen Faserdurchmesser von max. 9µm (ein menschliches Haar hat ca. 100µm Durchmesser) ist die Verarbeitung von Monomodefasern deutlich aufwändiger als bei Multimode-LWL.

Kunststofflichtwellenleiter
In der Netzwerktechnik findet man Kunststofflichtwellenleiter eher selten. Der Vollständigkeit halber hier trotzdem ein paar Worte zu dieser Technik.

Für die Datenübertragung mittels Kunststoff-LWL werden meist Polymerfasern verwendet. Mit einem gebräuchlichen Durchmesser von 1mm ist es auf jeden Fall eine

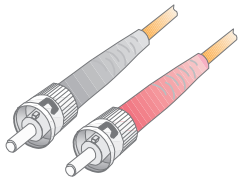
Multimodefaser. Die hohe Dämpfung des Polymermaterials beschränkt die maximale Länge der Übertragungsstrecken auf 20 - 100m. Der Haupteinsatz für Kunststoff-LWL ist deshalb die Übertragung von seriellen Signalen wie z.B. RS232 oder RS422/485.

Steckertypen

Eine weitere Varianz gibt es bei den LWL-Steckverbindungen. Hier gibt es drei grundsätzliche Verfahren:

Steckverbindungen mit Bajonet-Verriegelung, Steckverbindungen mit Überwurfmutter und Push/Pull-Steckverbinder mit Federarretierung

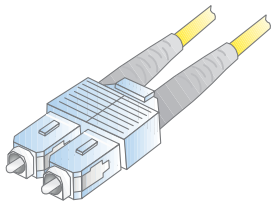
.



ST-Stecker

LWL-Type:	Multimode, Monomode
Verrieglung:	Bajonett-Verschluss
Einsatzgebiet:	LAN, WAN, serielle Signale

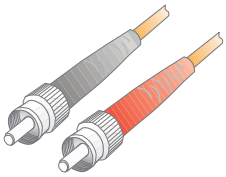
Über lange Zeit war die ST-Steckverbindung im Netzwerk- und Industriebereich die meist genutzte. Verdrehschutz und Bajonetverschluss verleihen dem ST-Stecker eine sichere und einfache Handhabung.



SC-Stecker

LWL-Type:	Multimode, Monomode
Verrieglung:	Push/Pull
Einsatzgebiet:	LAN, WAN, serielle Signale

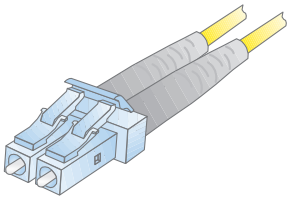
Bedingt durch seine einfache Push/Pull-Handhabung und die Duplexfähigkeit hat der SC-Steckverbinder heute die ST-Technik als meist verbreitetste abgelöst.



SMA-Stecker

LWL-Type:	Multimode, Monomode
Verrieglung:	Überwurfmutter
Einsatzgebiet:	LAN

Der SMA-Steckverbinder wurde in den Anfangszeiten der LWL-Technik eingesetzt. Der fehlende Verdrehschutz und die Gefahr des zu festen Anziehens führten oft zu Beschädigungen der Faser, weshalb die SMA-Technik heute kaum noch Bedeutung hat.



LC-Stecker

LWL-Type:	Multimode, Monomode
Verrieglung:	Push/Pull
Einsatzgebiet:	LAN, WAN

Wegen seiner kompakten Bauform wird der LC-Steckverbinder vorwiegend zur Konnektierung an Switches und anderen aktiven Netzwerkkomponenten eingesetzt.

An dieser Stelle haben wir nur die vier meist genutzten Stecksysteme vorgestellt. Eine vollständige Liste aller LWL-Steckervarianten würde den Rahmen sprengen.

LWL-Standards

Neben den unterschiedlichen Steckverbindern gibt es bei Ethernet über Glasfaser verschiedene Standards, die sich insbesondere hinsichtlich der Datenübertragungsraten unterscheiden.

Hier ein kurzer Überblick:

	Datenrate	LWL-Typ	Wellenlänge	Faser	Faserzahl	Max. Länge	Stecker	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

	Datenrate	LWL-Typ	Wellenlänge	Faser	Faserzahl	Max. Länge	Stecker	IEEE-
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	-
1000Base-ZX/-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
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

MAC-ID

Die unveränderbare physikalische Adresse einer Netzwerkkomponente. MAC = Media Access Control. *(Siehe. a. Ethernet-Adresse)*

Monomode-LWL

siehe LWL

Multimode-LWL

siehe LWL

NAT – Network Address Translation

Durch die explosionsartige Ausweitung des Internets in den letzten Jahren sind freie IP-Adressen knapp geworden und werden nur noch sehr sparsam vergeben. NAT

kommt dort zum Einsatz, wo Firmennetze ans Internet angebunden werden. Das Firmennetz ist über einen NAT-fähigen Router mit dem Internet verbunden, arbeitet intern allerdings mit einem eigenen, vom Internet unabhängigen IP-Adressraum. Von außen ist das Netz nur über eine einzige (oder einige wenige) IP-Adresse(n) ansprechbar. Anhand der Portnummer im empfangenen TCP/IP-Paket wird dieses an einen bestimmten internen Netzteilnehmer weitergeroutet.

Netzwerkknoten

Alle ans Netzwerk angeschlossenen Endgeräte kann man grundsätzlich auch als Netzwerkknoten bezeichnen.

NTBA

Network Termination for ISDN Basic rate Access, kurz NTBA, war der Leitungsabschluss bei ISDN-Anschlüssen.

Ping – Packet Internet Gopher

Ping dient in TCP/IP-Netzen zu Diagnosezwecken; mit Hilfe dieser Funktion lässt sich überprüfen, ob ein bestimmter Teilnehmer im Netz existiert und tatsächlich ansprechbar ist.

Die von Ping verwendeten ICMP-Pakete sind im Internet-Standard RFC-792 definiert.
(*Siehe. a.* 73)

PoE - Power over Ethernet

Mit PoE können Ethernet-Endgeräte die Versorgungsenergie aus dem Netzwerkkabel beziehen und so auf eine zusätzliche Stromversorgung verzichten. Die Versorgungsspannung wird von speziellen PoE-Switches oder speziellen Zwischenadaptern in das Netzwerkkabel eingespeist. (*Siehe. a.* 16)

POP3 – Post Office Protocol Version 3

Um eingegangene E-Mails aus dem Postfach auf dem Mailserver abzuholen, wird in den meisten Fällen das POP3-Protokoll benutzt. POP3 setzt auf TCP auf.
(*Siehe. a.* 107)

Port

Unter TCP und UDP bestimmt die Portnummer, an welche Anwendung ein eingehendes Datenpaket weitergereicht wird. (*Siehe. a.* 29)

PPP – Point to Point Protocol

PPP ist ein erweiterter Nachfolger von SLIP und weist u.a. eine verbesserte Fehlerkorrektur auf.

Genau wie SLIP bietet PPP die Möglichkeit, TCP/IP-Geräte, die keinen LAN-Anschluss haben, über die serielle Schnittstelle in TCP/IP-Netze einzubinden.
(*Siehe. a.* 56)

PPS-System - Produktionsplanungs- u. Steuerungssystem
Software-Lösung zur Produktionsplanung mit dem Ziel Produktionszeiten zu verkürzen, Bestands- und Lagermengen zu optimieren und Termine einzuhalten. Bekanntester Anbieter von PPS-Systemen ist SAP.

PPTP - Point-to-Point Tunneling Protocol
PPTP wurde ursprünglich von Microsoft, 3COM und anderen Firmen entwickelt, um PCs über öffentliche Netzwerke netztwerktechnisch mit Servern zu verbinden. Da PPTP Bestandteil von Windows Betriebssystemen ist, wird es auch heute noch für den Aufbau von VPN genutzt. (*Siehe. a.* 190)

Proxy (-Server)
Als Proxy bezeichnet man einen Server, der Inhalte von Webseiten zwischenspeichert. Ist ein Proxy vorhanden, fordert der Browser die gewünschte Webseite unter Angabe der eigentlichen URL beim Proxy an. Sind die Inhalte dort bereits zwischengespeichert, werden sie nicht erneut aus dem Internet geladen, sondern aus dem internen Speicher genommen und an den Browser weitergegeben. Das reduziert den Datenverkehr zum Internet. Vom Proxy gespeicherte Inhalte werden nach einer bestimmten Zeit wieder gelöscht bzw. neu geladen, damit der Datenbestand aktuell bleibt. (*Siehe. a.* 100)

Repeater
In 10Base2-Netzen dienen Repeater zur Verbindung zweier Ethernet-Segmente, um das Netz über die Ausdehnung eines einzelnen Segmentes hinaus zu erweitern. Repeater geben Datenpakete von einem Netzwerksegment zum anderen weiter, indem sie zwar die elektrischen Signale normgerecht „auffrischen", den Inhalt der Datenpakete dabei aber unverändert lassen. Erkennt der Repeater auf einem der angeschlossenen Segmente einen physikalischen Fehler, wird die Verbindung zu diesem Segment abgetrennt („partitioniert"). Die Partitionierung wird automatisch aufgehoben, wenn der Fehler nicht mehr vorhanden ist.

Zwischen zwei Stationen dürfen nicht mehr als vier Repeater liegen. Diese Regel betrifft allerdings lediglich „hintereinander" liegende Repeater – bei der Realisierung baumartiger Netzwerkstrukturen kann also durchaus eine Vielzahl von Repeatern eingesetzt werden.

RIP – Routing Information Protocol

Routingprotokolle wie RIP dienen dazu, Veränderungen der Routen zwischen zwei vernetzten Systemen an die beteiligten Systeme weiterzuleiten und so eine dynamische Änderung der Routingtabellen zu ermöglichen. RIP ist im Internet-Standard RFC-1058 definiert.

Router

Router verbinden zwei unterschiedliche Netze, wobei im Gegensatz zu Bridges nicht anhand der Ethernet-Adresse, sondern in Abhängigkeit von der IP-Adresse entschieden wird, welche Datenpakete wohin weiterzuleiten sind. (*Siehe. a.* 39)

Singlemode-LWL

(Siehe LWL)

SLIP – Serial Line Internet Protocol

SLIP bietet eine einfache Möglichkeit zur Übertragung von TCP/IP-Datenpaketen über serielle Punkt-zu-Punkt-Verbindungen. Damit können Endgeräte, die nicht über einen LAN-Anschluss verfügen, auch über die serielle Schnittstelle ins Netzwerk eingebunden werden.

SLIP arbeitet nach einem sehr einfachen Algorithmus ohne eigene Datensicherungsverfahren: Dem eigentlichen IP-Datenpaket wird ein Startzeichen (dezimal 192) vorangestellt und ein Endzeichen (ebenfalls dezimal 192) angehängt. Um die binäre Transparenz zu erhalten, werden im Datenpaket vorkommende Start- und Endzeichen zuvor durch andere Sequenzen ersetzt. SLIP ist in RFC 1055 beschrieben.
(*Siehe. a.* 55)

SLIP-Router

Ein SLIP-Router stellt die Hardware und Funktionalität zur Verfügung, um serielle Endgeräte, die über einen TCP/IP-Stack verfügen, in ein Netztwerk einzubinden.

Com-Server stellen z.B. SLIP-Routing als Betriebsart zur Verfügung.

SMTP – Simple Mail Transfer Protocol

SMTP regelt den Versand von E-Mails vom Mailclient zum Mailserver (SMTP-Server) und zwischen den Mailservern und setzt auf TCP auf. (*Siehe. a.* 106)

SNMP – Simple Network Management Protocol

SNMP setzt auf UDP auf und ermöglicht die zentrale Administration und Überwachung von Netzwerkkomponenten.

SNMP ist in folgenden Standards spezifiziert: RFC 1052, RFC 1155, RFC 1156, RFC 1157, RFC 1213 und RFC 1441. (*Siehe. a.* 📄 *85)*

STP – Shielded Twisted Pair

Abgeschirmtes Datenkabel, bei dem jeweils 2 Kabeladern miteinander verdrillt sind. *(Siehe. a. Twisted Pair)*

Subnet-Mask

32-Bit-Wert, der festlegt, welcher Teil der IP-Adresse das Netzwerk und welcher den Netzwerkteilnehmer adressiert. (*Siehe. a.* 📄 *37)*

Switch

Ein Switch bietet wie ein Hub die Möglichkeit, mehrere Netzteilnehmer sternförmig miteinander zu verbinden. Switches vereinigen die Funktionalität eines Hub mit denen einer Bridge: Ein Switch „lernt" die Ethernet-Adresse des an einem Port angeschlossenen Netzteilnehmers und leitet dorthin nur noch diejenigen Datenpakete weiter, die an diesen Netzteilnehmer adressiert sind. Eine Ausnahme bilden dabei Broadcast-Meldungen, die an alle Ports weitergegeben werden (hier unterscheidet sich der Switch in seiner Funktion von einer Bridge, die Broadcast-Meldungen generell nicht weitergibt). (*Siehe. a.* 📄 *15)*

TCP – Transmission Control Protocol

TCP setzt auf IP auf und sorgt nicht nur für die Verbindung der Teilnehmer während der Datenübertragung, sondern stellt auch die korrekte Zustellung der Daten und die richtige Abfolge der Datenpakete sicher. (*Siehe. a.* 📄 *28)*

TCP/IP-Stack

Teil des Betriebssystems oder ein auf das Betriebssystem aufgesetzter Treiber, der alle für die Unterstützung des IP-Protokolls benötigten Funktionen und Treiber zur Verfügung stellt.

Telnet – Terminal over Network

In der Vergangenheit kam Telnet vor allem für den Fernzugriff über das Netzwerk auf UNIX-Servern zum Einsatz. Über eine Telnet-Anwendung (Telnet-Client) kann von einem beliebigen Rechner im Netz ein Fernzugriff auf einen anderen Rechner (Telnet-Server) erfolgen. Heute wird Telnet auch zur Konfiguration von Netzwerkkomponenten wie z.B. Com-Servern benutzt. Telnet wird unter TCP/IP normalerweise über Portnummer 23 angesprochen; für spezielle Anwendungen können aber auch andere Portnummern verwendet werden. Telnet setzt auf TCP/IP als Übertragungs- und Sicherungsprotokoll auf.

Telnet ist im Internet-Standard RFC 854 definiert. *(Siehe. a. 75)*

TFTP – Trivial File Transfer Protocol

Das Trivial File Transfer Protocol (TFTP) ist neben FTP ein weiteres Protokoll zur Übertragung ganzer Dateien. TFTP bietet nur ein Minimum an Kommandos, unterstützt keine aufwändigen Sicherheitsmechanismen und benutzt UDP als Übertragungsprotokoll. Da UDP ein ungesichertes Protokoll ist, wurden in TFTP eigene minimale Sicherungsmechanismen implementiert. *(Siehe. a. 83)*

Das Trivial File Transfer Protocol ist in den RFC-Standards 783, 906, 1350 und 1782 bis 1785 beschrieben.

Transceiver

Das Wort Transceiver ist eine Zusammensetzung aus Transmitter (Sender) und Receiver (Empfänger). Der Transceiver realisiert den physikalischen Netzzugang einer Station an das Ethernet und ist bei den modernen Ethernet-Topologien 10Base2 und 10BaseT auf der Netzwerkkarte integriert. Nur bei 10Base5 *(siehe. auch AUI-Anschluss)* ist der Transceiver als externe Komponente direkt am Netzwerkkabel angebracht.

Treiber

Software um Hardware-Komponenten oder Peripheriegeräte in ein Betriebssystem zu integrieren / einzubinden.

Twisted Pair

Datenkabel, bei dem jeweils zwei Kabeladern miteinander verdrillt sind. Hierdurch wird ein deutlich reduziertes Übersprechverhalten zwischen den Doppeladern in einem Kabel erreicht. Man unterscheidet bei Twisted-Pair-Kabeln zwischen ungeschirmten UTP-Kabeln (Unshielded Twisted Pair) und geschirmten STP-Kabeln (Shielded Twisted Pair).

TP-Kabel werden vor allem in der Netzwerktechnik eingesetzt und sind nach ihren maximalen Übertragungsfrequenzen kategorisiert; in der Praxis kommen heute meist zwei Typen zum Einsatz:

- Kategorie-3-Kabel erlauben eine maximale Übertragungsfrequenz von 25MHz, ausreichend für den Einsatz in 10BaseT-, aber auch 100BaseT4-Netzen.
- Kategorie-5-Kabel erlauben eine maximale Übertragungsfrequenz von 100MHz und reichen damit für alle heutigen Netzwerktopologien aus.

UDP – User Datagram Protocol

UDP ist ein Protokoll, das wie TCP auf IP aufsetzt, im Gegensatz dazu aber verbindungslos arbeitet und über keine Sicherheitsmechanismen verfügt. Der Vorteil von UDP gegenüber TCP ist die höhere Übertragungsgeschwindigkeit. (*Siehe. a.* 📄 *31)*

UMTS - Universal Mobile Telecommunications System

Mit UMTS entstand nach der analogen Mobiltelefonie und GSM die dritte Generation der Mobilfunktechnik. Bei UMTS steht nicht mehr die Telefonie im Vordergrund. Vielmehr wurde UMTS bereits bei der Entwicklung auf die Nutzung vielfältiger multimedialer Dienste ausgerichtet.

UMTS-Endgeräte senden über ein Frequenzband von 1920MHz bis 1980MHz und empfangen bei 2110MHz bis 2170MHz. Die benutzbaren Einzelfrequenzen liegen jeweils 5MHz auseinander. Auf einer Einzelfrequenz können mehrere hundert Kanäle betrieben werden. Diese gleichzeitige Nutzung einer Frequenz wird nicht wie bei GSM über feste zeitliche Zuordnung geregelt. Bei UMTS regelt ein spezielles Protokoll die Nutzung. So können wenige Nutzer auf einer Frequenz große Datenmengen übertragen oder die Frequenz kann von vielen Nutzern zur Übertragung geringerer Datenmengen in Anspruch genommen werden.

Auf diese Weise lassen sich Übertragungsraten von bis zu 384kbit/s erreichen.
(*Siehe. a.* 📄 *205)*

URL – Uniform Resource Locator

Adress- und Protokollinformation für den Browser. Über die URL gibt der Anwender dem Browser vor, welches Protokoll genutzt wird, auf welchem Webserver die Seite liegt, und wo diese auf dem Webserver zu finden ist. (*Siehe. a.* 📄 *209)*

UTP – Unshielded Twisted Pair

Im Gegensatz zu Shielded Twisted Pair ein nicht abgeschirmtes Datenkabel, bei dem jeweils zwei Kabeladern miteinander verdrillt sind.

VPN - Virtual Private Network

VPN beschreibt die Technik, vertrauliche Netzwerkteile an verschiedenen Standorten, über das Internet, also ein öffentliches Netz, miteinander zu verbinden.

(*Siehe. a.* 📄 *184)*

Web-Based Management

Unter Web-Based Management versteht man die Möglichkeit, ohne spezielle Software Endgeräte über das Netzwerk direkt im Browserfenster zu konfigurieren.

Web-IO

Kleine Boxen mit Ethernet-Anschluss und integriertem Webserver. Web-IO können digitale oder analoge Signale über TCP/IP-Ethernet zugänglich machen bzw. im Browser visualisieren bzw. steuerbar machen.
(Siehe https://www.wut.de/web-io)

Wireless LAN

WLAN realisiert die Netzwerkanbindung über Funk. (*Siehe. a.* *19)*

WWW – World Wide Web

WWW wird häufig mit dem Internet gleichgesetzt. Das stimmt nicht ganz: Während das Internet die physikalischen Verbindungswege beschreibt, definiert das WWW die Gesamtheit der über das Internet verlinkten Webseiten bzw. Dokumente, die über das HTTP-Protokoll vom Browser geladen und angezeigt werden können.

Zahlensysteme

Wenn wir im Alltag mit Zahlen zu tun haben, dann sind das in aller Regel Dezimalzahlen. Das dezimale Zahlensystem ist uns vertraut und bereits als Kind lernt jeder, dass nach der Neun die Zehn kommt und die aufgeschriebene Zahl damit eine Stelle mehr hat.

Computer pflegen einen anderen Umgang mit Zahlen als Menschen und deshalb wollen wir hier ein wenig die Hintergründe von Zahlensystemen beleuchten.

Wert und Darstellung

Als Mensch hat man es täglich mit Zahlen und Berechnungen zu tun - aufgeschrieben in einem Rechenheft, als Anzahl von Münzen in der Tasche, aufgedruckt auf Verkehrsschildern usw.

Das menschliche Gehirn verarbeitet das alles als Zahlenwerte und kann sofort damit rechnen.

Computer unterscheiden hingegen streng zwischen Zahlenwerten, mit denen gerechnet werden kann und Zahlen bzw. Ziffern, die z.B auf einem Bildschirm dargestellt werden.

Wer sich schon einmal mit Programmiersprachen beschäftigt hat, weiß, dass „aufgeschriebene“ Zahlen vom Computer zunächst in Werte konvertiert werden müssen, bevor er damit rechnen kann.

In der Computertechnik haben wir es zudem mit unterschiedlichen Zahlensystemen zu tun.

- Dezimales Zahlensystem
- Duales/Binäres Zahlensystem
- Hexadezimales Zahlensystem

Wenn wir von verschiedenen Zahlensystemen reden, dann reden wir auch davon, dass ein und derselbe Wert auf verschiedene Weise dargestellt bzw. aufgeschrieben sein kann.

Das Dezimalsystem

Um andere Zahlensysteme zu verstehen, ist es wichtig, zunächst zu vergegenwärtigen, warum das dezimale Zahlensystem so ist, wie es ist.

Das Dezimalsystem fußt auf der Basis 10, was daran liegt, dass der Mensch zehn Finger hat (mit denen man prima abzählen kann).

Zur Darstellung dezimaler Zahlen steht ein Zeichenvorrat von 0 bis 9, also 10 Zeichen zur Verfügung.

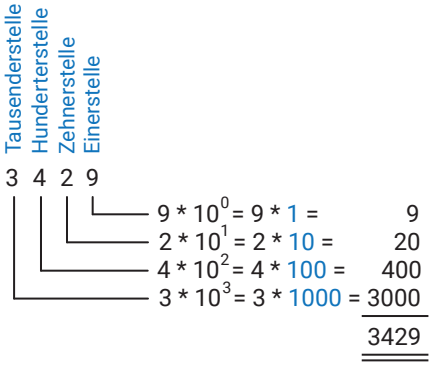
Zahlen, die größer sind als neun, erhalten eine neue Stelle, die Zehnerstelle. Bei Zahlen größer 99 kommt die Hunderterstelle dazu.

Schauen wir uns an einem Beispiel einmal die Systematik an, die dahinter steckt:

Mathematisch gesehen steht jede Stelle für die entsprechende Zehnerpotenz multipliziert mit der Ziffer, die an der Stelle steht.

Zur Erinnerung an den Mathematikunterricht:

- jede Zahl hoch 0 ist gleich 1
- jede Zahl hoch 1 ist die Zahl selbst
- jede Zahl hoch 2 ist die Zahl multipliziert mit sich selbst
- jede Zahl hoch 3 ist die Zahl multipliziert mit sich selbst und das Ergebnis nochmal multipliziert mit der Zahl
- usw.



Das duale/binäre Zahlensystem

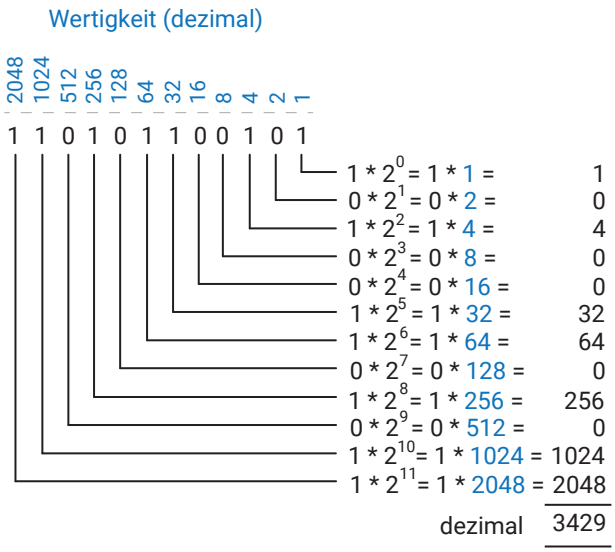
Das duale Zahlensystem, häufig auch als binäres Zahlensystem bezeichnet, ist das Zahlensystem, mit dem Mikroprozessoren und somit Computer intern arbeiten. Es kennt nur zwei Ziffern, nämlich die 0 und die 1 zur Darstellung einer Zahl.

Daraus resultiert, dass Zahlen, die größer sind als 1, bereits eine weitere Stelle bekommen. Zählt man duale Zahlen hoch, ergibt sich folgendes:

```
  0
  1
 10
 11
100
...
```

Die Systematik dahinter ist die gleiche wie beim dezimalen Zahlensystem, nur dass sie anstelle von Zehnerpotenzen auf Zweierpotenzen aufsetzt.

Unsere Zahl 3429 sieht dual geschrieben so aus: 110101100101



Aber warum arbeiten Mikroprozessoren eigentlich mit dualen Zahlen? Das hängt damit zusammen, dass Mikroprozessoren auf unterster Ebene nur zwei Zustände kennen: ON und OFF bzw. 1 und 0. Eine Stelle im dualen System entspricht einem Bit.

Das hexadezimale Zahlensystem

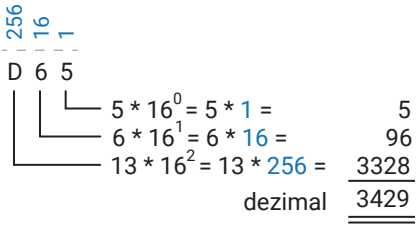
Beim hexadezimalen Zahlensystem gibt es innerhalb einer Stelle 16 mögliche Ziffern bzw. Zeichen. Da es als Ziffern nur 0 bis 9 gibt, werden Werte größer 9 mit den Buchstaben A bis F dargestellt (A=10, B=11, C=12, D=13, E=14 und F=15).

Zählt man hexadezimale Zahlen hoch, sieht es so aus:

```
 .
 8
 9
 A
 B
 C
 D
 E
 F
10
11
..
```

Und wieder ist die Systematik dieselbe wie beim dezimalen und dualen Zahlensystem. Allerdings wird beim hexadezimalen Zahlensystem mit Potenzen zur Basis 16 gearbeitet:

Wertigkeit (dezimal)



Nun kann man sich zu Recht fragen: Wozu braucht es ein weiteres Zahlensystem, das noch komplizierter und unüberschaubarer als das duale anmutet?

Wie bereits erklärt, arbeiten Computer mit Bits und Bytes - stellenweise sogar mit 16-Bit, 32-Bit oder 64-Bit-Werten.

Von der Logik her eignet sich zur Darstellung solcher Werte am besten das duale Zahlensystem, da jedes Bit durch eine Stelle repräsentiert wird. Für den Menschen sind Kolonnen von Einsen und Nullen aber schwer überschaubar.

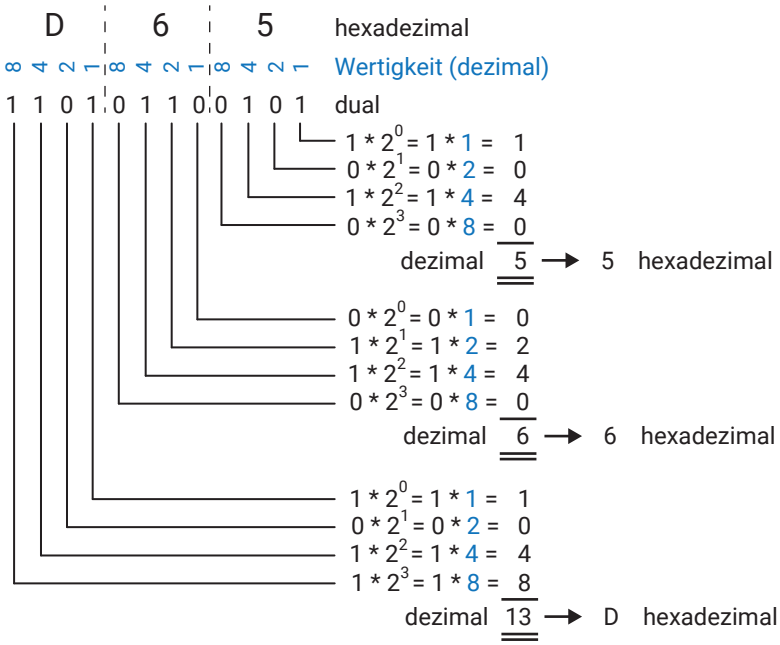
Sollen zum Beispiel ein paar 16-Bit-Werte eingegeben werden, ist die Wahrscheinlichkeit des Vertippens recht hoch.

Nun könnte man die Werte natürlich in dezimaler Schreibweise eingeben, aber dabei geht jeder Bezug zu dem Bit-Muster der Dualzahl verloren.

Dass 3429 dezimal = 110101100101 dual ist, ist ohne aufwändige Umrechnung nicht erkennbar. Und hier kommt das hexadezimale Zahlensystem ins Spiel.

Dass vier Stellen dual immer genau einer Stelle hexadezimal entsprechen, ist kein Zufall. Das hexadezimale Zahlensystem arbeitet mit Potenzen zur Basis 16 und 16 ist wiederum das Ergebnis einer Zweierpotenz, nämlich 2^4.

Es müssen also jeweils immer nur vier Stellen dual - also vier Bit - umgerechnet werden.



Mit ein bisschen Übung lassen sich so duale Zahlen im Kopf in hexadezimale umrechnen und umgekehrt.

Index

D

E

F

G

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

TCP/IP-Ethernet

Mach es einfach - Netzwerktechnische Grundlagen

Dieses Grundlagenbuch bietet einen schnellen Einstieg
in die Netzwerktechnik und nützliches Basiswissen.

Von der physikalischen Übertragung über logische Adressierung, Datentransport bis hin zur Netzwerksicherheit - hier erhalten Sie einen Überblick über die aktuellen Standards und ihre Hintergründe.

Anschauliche Grafiken und Beispiele helfen,
das Gelesene zu vertiefen.

Das sagen unsere Leser dazu:

"Eine so schöne Zusammenfassung und grafische Darstellung vieler Szenarien habe ich so noch nicht gesehen. Ich komme gar nicht mehr davon ab, immer weiter zu lesen."
(Mark H., Fachinformatiker & Geschäftsführer)

"Es gibt wenige Handbücher, die mich ein ganzes Arbeitsleben begleitet haben. Das neue Buch ist wieder einmal eine hervorragende Zusammenstellung und schließt bei mir einige Wissenslücken. Mir ist kein Fachbuch bekannt, das dieses Thema so strukturiert und einfach darstellt."
(Frank W., Infrastructure, Safety & Environment)

Wiesemann & Theis GmbH
Porschestraße 12
D-42279 Wuppertal

Mail info@wut.de
Web www.wut.de

Tel. +49 (0)202 2680-110
Fax +49 (0)202 2680-265