Einführung in TCP/IP

Bevor wir uns mit den eigentlichen Grundlagen zu TCP/IP befassen, wenden wir zunächst unsere Aufmerksamkeit den historischen Ursprung von TCP/IP zu. TCP/IP ist Teil der Internetprotokollfamilie (engl. Internet Protocol Suite) und wurde erstmalig Mitte der 70er Jahre entwickelt, als bei der amerikanischen DARPA das Interesse an einem Paketvermittlungsnetz aufkam, das die Kommunikation zwischen unterschiedlichen Computersystemen an Forschungseinrichtungen erleichtern sollte.

TCP/IP stellt die Basis für die Kommunikation im Internet dar indem es ein einheitliches Netzwerk mit offenen Protokollen bereitstellt, so dass unabhängig von der Hardware oder vom Ihrem Betriebssystem, kommuniziert werden kann. Mittlerweile hat sich das TCP/IP als Schlagwort für zwei ganz bestimmte Protokolle etabliert, obwohl es eigentlich eine ganze Familie von rund 500 Netzprotokollen repräsentiert. Dennoch gilt TCP/IP heute als gängiges Synonym für das TCP und das IP. Als TCP/IP Ende der 70er Jahre dem BSD-Unix beigefügt wurde, entwickelte sich daraus die Grundlage, auf der das Internet basiert.

Protokolle

Sobald Sie mit Netzwerken zu tun haben, müssen Sie sich mit Protokollen beschäftigen. Ein Netzwerkprotokoll (auch Netzprotokoll, Übertragungsprotokoll) ist eine exakte Vereinbarung (Protokoll), nach der Daten zwischen Computern bzw. Prozessen ausgetauscht werden, die durch ein Netzwerk miteinander verbunden sind. Die Vereinbarung besteht aus einem Satz von Regeln und Formaten (Syntax), die das Kommunikationsverhalten der kommunizierenden Instanzen in den Computern bestimmen (Semantik). Das ist vergleichbar mit natürlichen Sprachen, die ebenfalls auf einer Grammatik beruhen. Eine Sprache verfügt über einen Wortschatz, welcher semantische Informationen enthält und eine Grammatik, welche die Wörter in Beziehung zueinander setzt. Wenn Sie mit einem Spanier kommunizieren wollen, müssen Sie in der Regel spanisch sprechen können.

Der Austausch von Nachrichten erfordert häufig ein Zusammenspiel verschiedener Protokolle, die unterschiedliche Aufgaben übernehmen (beispielsweise Internetprotokollfamilie). Um die damit verbundene Komplexität beherrschen zu können, werden die einzelnen Protokolle in Schichten organisiert. Im Rahmen einer solchen Architektur gehört jedes Protokoll einer bestimmten Schicht an und ist für die Erledigung der speziellen Aufgaben zuständig (beispielsweise Überprüfen der Daten auf Vollständigkeit – Schicht 2). Protokolle höherer Schichten verwenden Dienste von Protokollen tieferer Schichten (Schicht 3 verlässt sich z. B. darauf, dass die Daten vollständig angekommen sind). Zusammen bilden die so strukturierten Protokolle einen Protokollstapel – in Anlehnung an das ISO-OSI-Referenzmodell (siehe auch DoD-Schichtenmodell). Nachrichten einer bestimmten Schicht werden auch als Protokolldateneinheiten bezeichnet.

Schichtenmodelle und Protokollarchitektur

Jede moderne Netzwerktechnik würde ohne Schichtenmodelle kaum funktionieren. Schichtenmodelle sind vereinfacht gesagt abstrakte Konstrukte um komplexe Sachverhalte einfach darstellen zu können. Der Vorteil eines schichtenmodellaufgebauten Netzwerkprotokolls liegt in seiner Flexibilität. Einzelne Schichten können durch eigene Protokolle angepasst bzw. erweitert werden ohne die Kommunikation der anderen Schichten untereinander zu gefährden. Bei der konzeptionellen Entwicklung eines Schichtenmodells ist also weniger die Geschwindigkeit oder das Leistungsverhalten entscheidend, da Schichtenmodelle aufgrund der komplexen Kommunikation der einzelnen Schichten untereinander diese Faktoren sowieso spürbar negativ beeinflussen können, als ein robuster Aufbau.

Die wesentlichen großen Schichtenmodelle in der Netzwerkwelt sind das *DoD-Schichtenmodell* und das
OSI-Schichtenmodell
. Während OSI sehr abstrakt und akademisch gehalten ist und die einzelnen Protokollschichten sehr detailliert implementiert, ist das DoD-Schichtenmodell näher an die Struktur der Protokolle angelehnt.

Das DoD-Schichtenmodell

Das *DoD-Schichtenmodell* ist ein theoretisches Modell des Department of Defense, kurz DoD (das US-Verteidigungsministerium), nach dem das Internet aufgebaut wurde. Es handelt sich um ein 4-Schichten-Netzwerkmodell das die einzelnen Aufgaben bei der Datenübertragung im Internet in aufeinander aufbauende Schichten einteilt. Basierend auf DoD wurde für das Internet das TCP/IP-Referenzmodell entwickelt.

[thumb src="images/tutorials/technic/dod-model.gif" arg=";;;Aufgabenverteilung in den einzelnen Schichten."]TCP/IP-Referenzmodell[/thumb]

Jede Schicht besteht aus einer Anzahl von Protokollen, die gemeinsam die TCP/IP-Protokollfamilie bilden. Die Spezifikationen für jedes Protokoll wurden damals jeweils in einem oder mehreren sogenannten RFC's festgelegt. Die Daten werden wie im nachfolgend beschriebenen OSI-Modell beim Versenden im Stack nach unten gereicht. Beim Empfang von Daten aus dem Netz führt der Weg durch den Stack nach oben. Jede Schicht fügt dabei ihre Kontrollinformationen hinzu, um eine korrekte Übertragung der Daten sicherzustellen. Diese Informationen nennt man *Header*, da diese den eigentlichen Daten vorangestellt werden.

Das OSI-Schichtenmodell

Beim *OSI-Schichtenmodell* (engl. Open Systems Interconnection Reference Model) handelt es sich um ein um drei Schichten erweitertes Modell zur Kommunikation informationsverarbeitender Systeme. Das Modell beschreibt vereinheitlichte Verfahren und Regeln für den Austausch von Daten und schafft einen Bezugsrahmen. Es wurde im Jahre 1979 entwickelt und von der ISO standardisiert.

Das OSI-Modell dient seitdem als Grundlage für eine Reihe von herstellerunabhängigen Netzprotokollen, die in der öffentlichen Kommunikationstechnik im Transportnetz fast ausschließlich eingesetzt werden. In Computernetzwerken werden den verschiedenen Hosts *Di enste* unterschiedlichster Art bereitgestellt und zwar von den anderen Teilnehmern im Netz. Dazu sind einige abstrakte technische Anforderungen zu bewältigen. Zusammenfassend lässt sich das OSI-Modell in 6 Punkten konkretisieren:

- Zur Beschreibung der Struktur und Funktion von Protokollen für die Datenkommunikation wird ein Architekturmodell zugrundegelegt, das von der International Standards Organisation (ISO) entwickelt wurde.
- Dieses Basisreferenzmodell namens *Open Systems Interconnect (OSI) Reference Model* schafft einen Bezugsrahmen für die Behandlung von Themen aus dem Bereich der Datenkommunikation.

- Das Basisreferenzmodell der ISO besteht aus 7 Schichten (layers). Diese Auteilung dient dazu die Probleme und Aufgaben auf individuelle Ebenen aufzuteilen auf derer die Instanzen die Anforderungen umsetzen.
- Jede dieser Schichten definiert gewisse Funktionen der Protokolle für die Datenkommunikation, die beim Austausch von Daten zwischen Anwendungen über ein dazwischenliegendes Netzwerk hinweg ausgeführt werden.
- Der reale Datenfluss erfolgt vertikal. Die Instanzen einer Schicht sind austauschbar, sofern sie sowohl bei Sender als auch Empfänger ausgetauscht werden.
- Jede einzelne Schicht definiert nicht ein Protokoll, sondern stellt vielmehr eine Funktion der Datenkommunikation dar, die von beliebig vielen Protokollen ausgeführt werden kann. Jede

Schicht kann mehrere Protokolle enthalten, von denen jedes solche Dienste bereitstellt, wie sie für die Erfüllung der Funktion dieser Schicht benötigt werden. Es ergibt sich das folgende Modell:

[thumb src="images/tutorials/technic/osi-model.jpg" arg=";;;Details der einzelnen Schichten."]OSI-Schichtenmodell[/thumb]

Das OSI-Modell mit den zugehörigen Protokollen und Einheiten im Überblick:

OSI-Schicht	Einordnung	DoD-Schicht	Einordnung
7 (Application) orientiert	Anwendungen Anwendungs- Anwendung	Ende zu	
Ende (Multihop) FTP HTTPS SMTP LDAP NCP	HTTP Daten	Gateway, Switch, Layer 4-7 Switch	
6 (Presentation)	Darstellung		
5 (Session)	Sitzung		
4 (Transport) orientiert	Transport Transport- Transport	TCP	
UDP SCTP SPX	Segmente		
3 (Network)	Vermittlung Internet	ICMP	
IGMP IP IPX	Pakete	Router, Layer-3-Switch	
2 (Data Link)	Sicherung Netzzugang		
Punkt zu Punkt Token Ring FDDI ARCNET	Ethernet Rahmen (Frames)	Bridge, Switch	

1	Bitübertragung	
(Physical)	Bits	Repeater, Hub

In der letzten Spalte sind die Kopplungselemente dargestellt, die auf der entsprechenden Ebene für die physikalische Übermittlung der Daten zuständig sind.

TCP/IP (Transmission Control Protocol / Internet Protocol) im Detail

TCP/IP ist sowohl im UNIX-Bereich als auch auf dem PC (DOS, Windows, etc.) der Standard-Protokoll-Stack für die Anbindung an das Internet. Für die Internet-Protokoll-Familie ist dabei das TCP/IP-Referenzmodell wie bereits erläutert maßgebend. Es beschreibt den Aufbau und das Zusammenwirken der Netzwerkprotokolle aus der Internet-Protokoll-Familie. Analog zum theoretischen DoD-Schichtenmodell gliedert es sie in
4
aufeinander aufbauende Schichten. Daher auch Protokoll-Stack (
protocol stack
).

Das TCP/IP-Referenzmodell ist auf die Internet-Protokolle zugeschnitten, die den Datenaustausch über die Grenzen lokaler Netzwerke hinaus ermöglichen („*Internetworking*“). Es wird weder der Zugriff auf ein Übertragungsmedium noch die Datenübertragungstechnik definiert. Vielmehr sind die Internet-Protokolle dafür zuständig, Datenpakete über mehrere Punkt-zu-Punkt-Verbindungen (
Hops
) weiterzuvermitteln und auf dieser Basis Verbindungen zwischen Netzwerkteilnehmern über mehrere Hops herzustellen. Das
Internet Protokoll
(IP) selbst stellt dabei die erste unabhängige Schicht der Internet-Protokoll-Familie dar.

TCP/IP Eigenschaften:

TCP/IP hat einige wesentliche Merkmale. Zum Einen sind die Protokollspezifikationen offen, also jedem frei zugänglich und somit herstellerunabhängig. Zum Anderen ist es unabhängig von einem bestimmten Netzwerkmedium. Desweiteren stellt es ein einheitliches Adressierungsschema bereit und verfügt über standardisierte Schnittstellen zu Anwendungsprogrammen.

[thumb src="images/tutorials/technic/protocol-architecture.gif" arg=";;;Die Protokollarchitektur

von TCP/IP im Vergleich zum OSI-Modell."]Vergleich der Protokollarchitektur[/thumb]

Die Daten wandern durch die einzelnen Schichten und bekommen in jeder tieferen Schicht einen neuen Header (Kopf) mit Kontrollinformationen hinzugefügt -> Encapsulation (Kapselung). Beim Datentransport von unten nach oben, werden diese Zusatzinformationen wieder entfernt.

[thumb src="images/tutorials/technic/osi-scheme.gif" arg=";;;Bezeichnung der Daten in den einzelnen Schichten."]Datenbezeichnung[/thumb] **Die Netzzugangsschicht**

- Übertragung von Daten in einem direkt angeschlossenen Netzwerk.
- Definiert, wie ein IP-Datagramm über das Netzwerk transportiert wird.
- Jeder physikalische Netzwerkstandard braucht sein eigenes Protokoll.
- Abbildung von IP-Adressen auf physikalische Netzadressen.
- Dokumentation : RFC826 und RFC894.

Die Internetschicht

- Das Internet Protokoll (IP) definiert Transport von Datagrammen.
- Definition von Datagrammen (kleinste Einheit für die Übertragung im Internet).
- Definition der Internet-Adressierung.
- Routing von Datagrammen zu fremden Rechnern.
- Keine eigene Fehlerkorrektur.
- Bereitstellung des Internet Control Message Protocols (ICMP) zur Versendung von Kontrollinformationen:
- Flußkontrolle
- Erkennung unerreichbarer Ziele
- Änderungen im Routing
- Statusabfrage bei fremden Rechnern

- Dokumentation : RFC791 (IP), RFC792 (ICMP)

Die Transportschicht

- Bereitstellung des Transmission Control Protocol (TCP) und des User Datagram Protocol (UDP).
- TCP bietet Fehlererkennung und Korrektur auf dem gesamten Übertragungsweg.
- UDP bietet eine verbindungslose Übertragung mit geringem Verwaltungsaufwand.

Die Anwendungsschicht

- Bereitstellung von Anwendungssoftware, wie Ping, Telnet, FTP, Email, News, WWW, etc...

Mehr zum Internet Protocol

Das Internet Protocol (IP) ist die Grundlage der Protokollfamilie TCP/IP und für die Weiterleitung der Daten zuständig. Generell hat es die Aufgabe, die Datenübertragung zwischen Netzwerken sicherzustellen. Dazu muss das Protokoll diverse Aufgaben übernehmen und diese als Dienst den höheren Schichten zur Verfügung stellen. Zu den Aufgaben des IP zählen:

- Datenpaketdienst
- Fragmentierung von Datenpaketen
- Wahl der Übertragungsparameter
- Adressfunktion
- Routing zwischen Netzwerken

Die Hauptaufgabe des IP ist die Ermittlung und Realisierung des optimalen Weges zwischen Sender und Empfänger für jedes Datenpaket. Verbindungsaufbau und Verbindungsabbau fallen nicht in den Zuständigkeitsbereich dieses Protokolls. Das Internet Protocol stellt keine gesicherte Verbindung
zur Verfügung und kann keine verlorenen Datenpakete erneut übertragen. Jedes
IP-Datenpaket
wird als unabhängiges Paket (Datagramm) durch das Netzwerk an den Empfänger übermittelt. Für die Netzwerktypen sind unterschiedliche Datenpaketlängen festgelegt. Die Größe eines Datenpakets hängt von mehreren Faktoren ab, wie Hardware- und Software-Beschränkungen. Ist ein Datenpaket wegen seiner Überlänge nicht als eine Einheit übertragbar, so muss es in kleinere Fragmente zerlegt werden. Die Pakete werden zwar in der richtigen Reihenfolge gesendet, kommen aber nicht notwendigerweise in derselben dort an. Da die Einzelpakete verschiedene Wege gehen können, sind zusätzliche Informationen erforderlich. Diese erlauben, den Zustand des ursprünglichen Datenpakets zu rekonstruieren. Jedes Datenpaket erhält daher bei der Übertragung einen IP-Header vorangestellt.

Das Datagramm

Im Internet werden Daten als kleine Pakete (Datagramme) verschickt, ein direkter Verbindungsaufbau zum Zielrechner findet nicht statt. Jede Information wird vor ihrer Reise mit Kontrollinformationen versehen und in ein sogenanntes Datagramm "verpackt":

[thumb src="images/tutorials/technic/datagram.gif" arg=";;;Prinzipieller Aufbau eines Datagramms."]Datagramm[/thumb]

- Jedes einzelne Datagramm durchquert das Netz unabhängig von allen anderen Datagrammen.
- Ein Datagramm ist ein Paketformat, dessen erste fünf oder sechs 32-Bit-Wörter Kontrollinformationen enthalten und als Header bezeichnet werden.

- Weil die Länge des Headers variieren kann (das sechste Header-Wort ist optional), ist ein Feld namens Internet Header Length (IHL) enthalten, in dem die Länge angegeben ist. Der Header enthält alle Informationen, die für die Zustellung des Datagramms notwendig sind.

- Das Internet-Protokoll transportiert Datagramme, indem es die Destination Address (Zieladresse) im fünften Wort des Headers liest. Diese Zieladresse ist die Standard-IP-Adresse mit einer Länge von 32 Bit.
- Wenn die Zieladresse zu einem Rechner im lokalen Netzwerk gehört, wird das Datagramm auf direktem Wege zugestellt. Andernfalls wird es an einen Router übergeben (es wird geroutet).

IP-Adressen

Nachdem wir eine Menge über das Schichtenmodell gelernt haben und auf die Protokoll-Details zu TCP/IP eingegangen sind, kommen wir nun zu den *IP-Adressen* und anschließend zu den
Ports
. Bevor Sie sich aber den nächsten Teil dieses Artikels zu Gemüte führen haben Sie nun die Möglichkeit sich zurück zu lehnen und sich die virtuelle Reise eines IP-Paketes per Video anzusehen.

Als quasi Beifahrer konnten Sie nun die Reise eines IP-Paketes quer durch das Internet verfolgen. Wir möchten nun einen genaueren Blick auf die IP-Adresse werfen die unser Paket im Video sicher zum Zielrechner geleitet hat.

Das Internet Protokoll überträgt Daten zwischen Rechnern in Form von Datagrammen, wobei jedes Datagramm an die Adresse im Internet weitergeleitet wird, die im Feld "Zieladresse" des Datagramm-Headers angegeben ist. Diese sogenannte IP-Adresse besteht aus zwei Teilen, der Netzadresse und der Adresse des Rechners (des Host) im Netz.

[thumb src="images/tutorials/technic/ip-addresses.gif" arg=";;;IP-Adressen werden in Klassen aufgeteilt, je nachdem, mit welcher Bitkombination die Adresse beginnt."]Aufteilung IP-Adressen[/thumb]

- Ist das erste Bit der Adresse 0, dann gehört die Adresse zu der Klasse A:
- Bit 0 bis 7 bestimmen das Netzwerk.
- Bit 8 bis 31 bestimmen den Rechner.
- insg. 126 Netze mit je 16777214 Rechnern möglich.

- Sind die ersten beiden Bit der Adresse 10, dann gehört die Adresse zu der Klasse B:

- Bit 0 bis 15 bestimmen das Netzwerk.
- Bit 16 bis 31 bestimmen den Rechner.
- insg. 16382 Netze mit je 65534 Rechnern möglich.

- Sind die ersten drei Bit der Adresse 110, dann gehört die Adresse zu der Klasse C:
- Bit 0 bis 23 bestimmen das Netzwerk.
- Bit 24 bis 31 bestimmen den Rechner.
- insg. 2097150 Netze mit je 254 Rechnern möglich.

Beispiel:

Die Ruhr-Universität Bochum hat die Internet-Nummer 134.147.xxx.xxx. Das ergibt in dualer Schreibweise: 10000110.10010011.xxxxxxxx.xxxxxxxx. Somit gehören alle Netzadressen der Universität zur Klasse B.

- IP-Adressen werden in der Regel als vier durch Punkte getrennte Zahlen geschrieben, wobei jede dieser Zahlen im Bereich von 0 bis 255 liegt (1 Byte).
- Im Gegensatz zu IPv4 (32 Bit) umfasst die IP-Adresse in IPv6 insgesamt 128 Bit.
- In allen Klassen gibt es besondere reservierte Adressen, die nicht frei vergeben werden dürfen. Dazu gehören z.B. die Rechnernummern 0 und 255.
- Eine IP-Adresse, in der alle Rechnerbits auf 0 stehen, identifiziert das Netzwerk selbst. Adressen in diesem Format werden in Routing-Tabellen verwendet, um komplette Netzwerke zu adressieren.
- Eine IP-Adresse, in der alle Rechnerbits auf 1 stehen, bezeichnet man als Broadcast-Adresse. Eine solche Adresse wird benutzt, um gleichzeitig jeden einzelnen Rechner in einem Netzwerk zu adressieren. Ein Datagramm mit dieser Adresse wird von jedem einzelnen Rechner im Netzwerk gelesen und ausgewertet.

- IP benutzt den Netzwerkanteil einer Adresse, um ein Datagramm durch die Netze zu routen. Die komplette Adresse einschließlich dem Rechneranteil wird innerhalb des Zielnetzes für die endgültige Zustellung verwendet.

Ports

Ohne Ports wäre eine Kommunikation über die im Internet üblichen Protokolle (TCP und UDP) nicht möglich. Portnummern zählen zu den grundlegenden Elementen beim Einsatz der

Protokolle TCP und UDP. Eine Portnummer ist 16 Bit groß und insgesamt stehen jeweils 65.535 verschiedene TCP- und UDP-Ports zur Verfügung.

Sind die Daten am Zielrechner angekommen, müssen sie an den richtigen Anwendungsprozess ausgeliefert werden. Doch welche Daten gehören zu welcher Anwendung? Hier kommt die Port-Nummer ins Spiel. Sie ermöglicht es die Daten auch der korrekten Anwendung zuzuordnen. Identifiziert also die IP-Adresse den Rechner im Netzwerk, so ordnet die Port-Nummer im Paket dieses einer bestimmten Anwendung zu. Um einen Überblick zu behalten und bestimmten Applikationen feste Nummern zuweisen zu können, hat man diese in *drei Gruppen*
unterteilt:

Well Known Ports: Bei diesem Typ handelt es sich um reservierte und standardisierte Portnummern zwischen 1 und 1023. Dies vereinfacht den Aufbau einer Verbindung, weil sowohl Absender und Empfänger bereits wissen, dass Daten für einen bestimmten Prozess an einen bestimmten Port gesendet werden müssen. So nutzen beispielsweise alle Telnet-Server den Port 23. Die Well Known Ports ermöglichen den Clients die Verbindung zu Servern, ohne dass eine weitere Konfiguration notwendig ist. Die Verwaltung dieser Ports übernimmt die Internet Assigned Numbers Authority (IANA). Eine Liste der aktuell vergebenen Portnummern finden Sie hier .

Bis 1992 bewegten sich die Well Known Ports im Bereich zwischen 1 und 255. Die Nebenstellen zwischen 256 und 1023 wurden für Unix-spezifische Dienste verwendet.

Registered Ports: Diese Ports im Bereich von 1024 bis 49.151 sind für Dienste vorgesehen, die üblicherweise auf bestimmten Nebenstellen laufen. Ein Beispiel hierfür ist der Port 3128, der von Proxy-Servern oft alternativ für das Hypertext Transport Protocol (HTTP) verwendet wird.

Dynamically Allocated Ports: Diese auch Ephemeral Ports genannten Nebenstellen werden stets dynamisch zugewiesen. Sie liegen im Bereich von 49.152 bis 65.535. Jeder Client kann diese Ports nutzen, solange die Kombination aus Transportprotokoll, IP-Adresse und Portnummer eindeutig ist. Wenn ein Prozess einen Port benötigt, fordert er diesen bei seinem Host an.

Bei der Konfiguration einer Firewall ist beispielsweise ein Grundwissen über Portnummern vonnöten. Wie Sie in dem Video sehen konnten entscheidet nämlich ein Paketfilter bei jedem Datenpaket anhand festgelegter Filterregeln, ob er es weiterleitet oder nicht. Dabei werden unter anderem Header-Informationen wie Absender- und Zielport ausgelesen. Um beispielsweise den FTP-Service abzublocken, sondern die Firewall alle Pakete aus, die im *Head er*
den Port 21 eingetragen haben.

TCP- und UDP-Header

Die wesentlichen Datenpakete im Internet sind das TCP-Paket und das UDP-Paket. Ein Paket besteht für gewöhnlich aus zwei Teilen – dem *Header* und der Nutzlast (Payload). Die Nutzlast stellt die eigentlichen Daten dar, während der Header eine Reihe an für den Transport notwendigen Informationen enthält. Darunter die Port-Nummer und eine Prüfsumme. Nachfolgend ist der schematische Aufbau eines TCP-Paketes samt Header abgebildet:

[thumb src="images/tutorials/technic/tcp-header.gif" arg=";;;Die 16 Bit lange 'Destination Port'-Nummer legt fest, für welche Applikation das Datenpaket bestimmt ist."]Aufbau des TCP-Header[/thumb]

Die IP-Protokollnummer steht in einem Byte im dritten Wort des Datagramm-Headers. Dieser Wert bestimmt die Übergabe an das jeweilige Protokoll in der Transportschicht, beispielsweise "6" für TCP oder "17" für UDP. Das Transportprotokoll muss nach Empfang die Daten an den richtigen Anwendungsprozess übergeben. Anwendungsprozesse werden anhand der 16 Bit langen Portnummer identifiziert, an die die Daten nach Empfang auf dem Zielrechner übergeben werden. Im ersten Wort jedes TCP- und UDP-Headers sind daher sowohl die "Source Port"-Nummer als auch die "Destination Port"-Nummer enthalten. Soll also eine Applikation unter einer bestimmten Portnummer erreichbar sein, teilt sie dies dem TCP/IP-Protokoll-Stack mit.

Im Gegensatz zu TCP ist das User Datagram Protocol (UDP) ein minimales, verbindungsloses Netzprotokoll. Es ist auf Flexibilität und Einfachheit ausgelegt. UDP Datenpakete können maximal 65535 Bytes lang sein, wovon der IP-Header und UDP-Header insgesamt mindestens 28 Bytes belegen. Der UDP-Header selbst besteht aus vier Headerfeldern. UDP-Datagramme haben daher maximal 65507 Bytes an Nutzdaten. Nachfolgend ist der schematische Aufbau des UDP-Paketes samt Header abgebildet:

[thumb src="images/tutorials/technic/udp-header.gif" arg=";;;Da UDP verbindungslos ist, ist der Quell-Port optional. Ebenso die Prüfsumme."]Aufbau des UDP-Header[/thumb] **Das IP-Packet**

Sowohl TCP als auch UDP sind Bestandteil der **Transportschicht** im OSI-Schichtenmodell. Um Daten über das Internet zu versenden müssen diese Pakete im Protokollstack nach unten, zur Vermittlungsschicht, gereicht werden um dort beispielsweise in einem IP-Paket *verpackt* zu werden. Das IP-Paket oder exakt Internet Protocol Datagram ist das Grundelement der Internet-Datenkommunikation. Es besteht immer aus zwei Teilen: den Kopfdaten, die Informationen über Quelle, Ziel (IP-Adresse), Status, Fragmentierung, etc. enthalten, und den Nutzdaten. Das TCP-Protokoll zum Beispiel befindet sich ausschließlich in den Nutzdaten des IP-Pakets – eine Schicht weiter oben im OSI-Modell.

In den Kopfdaten stehen die ausschließlich protokollrelevanten Informationen eines IP-Pakets. Genau wie der Rest des gesamten Internet Protocol ist der Aufbau des Kopfdatenbereiches in der verbreiteten Version 4 des Protokolls (IPv4) im RFC 791 festgelegt. Nachfolgend sehen Sie den schematischen Aufbau eines IP-Packets.

IPv4 0 4 8 12 16 20 24 32 bit			
Version	IHL	TOS	Total Length
Identification	Flags	Fragment Offset	
Time to Live	Protocol	Header Checksum	
Source Address			
Destination Address			
Options and Padding			

Das neuere Protokoll Version 6 (IPv6) hat einen anderen Kopfdatenbereich.

IPv6 0 4 8 12 16 20 24 28 31 bit		
Version	Traffic Class	Flow Label
Payload Length	Next Header	Hop Limit
Source Address (128 Bit)		
Destination Address (128 Bit)		

Socket

Die Kombination aus IP-Adresse und Portnummer bezeichnet man als *Socket*! Sockets sind in der Netzwerkprogrammierung von wesentlicher Bedeutung. Mit Sockets ist es möglich, einen einzelnen Netzwerkprozess innerhalb des gesamten Internets eindeutig zu identifizieren. Die Notation ist folgende: IP-Adresse:Port, zum Beispiel 62.97.226.75:80. Zwei Sockets definieren eine Verbindung - einer für den Ausgangs- und einer für den Zielrechner.

Die exakte Definition eines Sockets lautet:

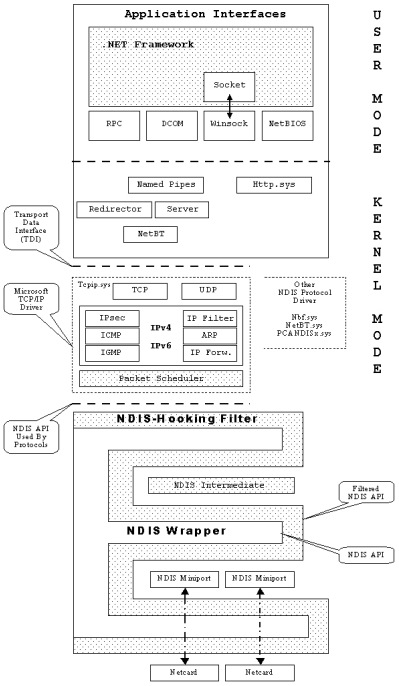
Definition: Ein Socket ist ein Endpunkt einer bi-direktionalen Software-Schnittstelle zur Interprozess- (IPC) oder Netzwerk-Kommunikation zwischen zwei Programmen. Ein Socket ist gebunden an eine Port-Nummer, so dass die TCP Schicht die Anwendung identifizieren kann für die die Informationen bestimmt sind.

Weitere Informationen dazu gibt es im C#-Tutorial TCP/IP Socket-Programmierung in C# .

TCP/IP-Stack Architektur in Windows

Jedes Betriebssystem verfügt über seine eigene TCP/IP-Stack-Implementation. Der TCP/IP-Stack hat die Aufgabe, eingehende und ausgehende IP-Pakete zu verarbeiten und die Paketdaten zur Verarbeitung an die entsprechende Anwendung weiterzuleiten. In Windows ist der TCP/IP-Stack in der Treiberdatei **tcpip.sys** realisiert. Die Windows Transmission Control Protocol/Internet Protocol (TCP/IP) Suite enthält Elemente der Kernprotokolle und Dienste, sowie Schnittstellen zwischen diesen.

Auf der untersten Ebene schließt dass das Transportprotokoll mit ein, welches als unabhängige Schnittstelle für die Kommunikation mit den Kernel-Mode Komponenten und Geräten fungiert. Zusätzlich stehen für die User-Mode Applikationen eine Reihe von High-Level Schnittstellen bereit. Die bekanntesten Schnittstellen sind Windows Sockets 2 (Winsock), Remote Procedure Call (RPC) und NetBIOS. Winsock dient unter Windows allen Programmen als Schnittstelle für den Zugriff auf ein Netzwerk mithilfe von Sockets. Der vollständige Aufbau der TCP/IP Suite ist nachfolgend abgebildet.



TCP in der Praxis

Nach den Grundlagen sehen wir uns nun einmal an, wie ein Verbindungsaufbau mithilfe von TCP in der Praxis aussieht. Das TCP stellt beim Verbindungsaufbau einen *Kanal* zwischen zwei Rechnern (genauer: Endpunkten zwischen 2 Anwendungen auf diesen Rechnern) her. Innerhalb dieses Kanal's können Daten in beide Richtungen übertragen werden. Der Verbindungsauf- und abbau gestaltet sich wie folgt:

Ein Server-Rechner, der einen Dienst wie beispielsweise elektronische Post anbietet, generiert einen Endpunkt mit einem fixen Port und seiner IP-Adresse (er kann auch beliebige Adressen zulassen). Dies wird als *PASSIVE OPEN* oder auch *LISTENING* bezeichnet.

Will ein Client eine Verbindung aufbauen, generiert er ebenfalls einen eigenen Endpunkt aus seiner IP-Adresse und einer noch freien Portnummer. Mit Hilfe des ihm bekannten Ports (z.B. Port 80) an welchem der Server seine Dienste anbietet und der IP-Adresse wird dann eine Verbindung aufgebaut. Für den Aufbau der Verbindung sind unter TCP drei Pakete erforderlich (3-Way-Handshake).

[thumb src="images/tutorials/technic/3-way-handshake.gif" arg=";;;Ablauf eines 3-Way-Handshakes."]3-Way-Handshake[/thumb]

Während der Datenübertragungsphase (active open) sind die Rollen von Client und Server (aus TCP-Sicht) vollkommen symmetrisch. Insbesondere kann jeder der beiden beteiligten Rechner einen Verbindungsabbau einleiten. Während des Abbaus kann die Gegenseite noch Daten

übertragen, die Verbindung kann also halb-offen sein. Ein 4-Wege-Handshake wird benutzt, um die Verbindung abzubauen.

Im Gegensatz zum paketorientierten UDP implementiert TCP einen bidirektionalen, byte-orientierten, zuverlässigen Datenstrom zwischen zwei Endpunkten. Das darunterliegende Protokoll (meist IP) ist paketorientiert, wobei Datenpakete verloren gehen können, in verkehrter Reihenfolge ankommen dürfen und sogar doppelt empfangen werden können. TCP prüft die Integrität der Daten mittels einer Prüfsumme und stellt die Reihenfolge durch Sequenznummern sicher. Der Sender wiederholt das Senden von Paketen falls keine Bestätigung innerhalb einer bestimmten Zeitspanne (Timeout) eintrifft. Die Daten der Pakete werden im Empfänger in einem Puffer zu einem Datenstrom zusammengefügt und doppelte Pakete verworfen.

Die jeweilige Länge des Puffers, bis zu der keine Lücke im Datenstrom existiert, wird bestätigt (Windowing). Dadurch ist die Ausnutzung der Netzwerk-Bandbreite auch bei großen Strecken möglich. Bei einer Übersee- oder Satellitenverbindung dauert das Eintreffen des ersten Acknowledges (ACK) aus technischen Gründen mehrere 100 ms, in dieser Zeit können unter Umständen mehrere hundert Pakete gesendet werden. Der Sender kann den Empfängerpuffer füllen bevor die erste Bestätigung eintrifft. Alle Pakete im Puffer können gemeinsam bestätigt werden. Bestätigungen werden zusätzlich zu den Daten in die Paket-Header im entgegengesetzen Datenstrom eingefügt (Piggybacking).

Querverweise

TCP/IP Socket-Programmierung in C#
Battleship
Transmission Control Protocol (Wikipedia)
The TCP/IP Guide