TCP / IP
Protokollfamilie

Historischer Überblick, Einführung

ARPAnet

ARPA ist die Forschungsinstitution des amerikanischen Verteidigungsministeriums (***Advanced Research Project Agency***). ARPAnet ist ***ältestes paketvermittelndes Netz auf TCP/IP-Basis***, zunächst auf Mietleitungen. Es wurde ***1969 unter der Federführung des amerikanischen Verteidigungsministeriums*** von BBN (Bolt, Beranek and Newman) erbaut und implementiert. Seiner Struktur wegen könnte man es als erstes Backbone-Netz bezeichnen, allerdings nur mit einer Datenübertragungsrate von 56 kbit/s. 1983 hatte das ARPAnet eine solche Ausdehnung erreicht, daß es in einen forschungsorientierten Teil (das ARPAnet) und einen militärischen Teil aufgeteilt wurde. Einige Jahre später wurden die Hauptübertragungswege von 56 kbit/s auf 1,5 Mbit/s aufgerüstet, und Anfang der 90er Jahre wurde das NSF-Backbone (National Science Foundation) auf 45 Mbit/s hochgerüstet. Seit 1993 wurden Netze mit ATM-Technologie erprobt, die heute Geschwindigkeiten bis 622 Mbit/s aufweisen.
Aufbauend auf dem ARPAnet wurden verschiedene Dienste definiert. Die ***ARPA-Dienste*** unterstützen die wichtigsten Anwendungen auf der Grundlage der TCP/IP-Protokolle. Sie sind in den Schichten 5 bis 7 angesiedelt und gelten sowohl für das TCP- als auch für das UDP-Protokoll:

Die DoD - Protokollfamilie

Das amerikanische Verteidigungsministerium (***Department of Defense, DoD***) hatte 1980 mangels standardisierter Protokolle eine Protokollfamilie für die offene Kommunikation ins Leben gerufen, die sogenannte DoD-Protokollfamilie. Die bekanntesten Protokolle dieser Familie sind das in OSI-Ebene 3 angesiedelte ***IP (internet protocol)*** und das in der Schicht 4 angesiedelte Transport-Protokoll ***TCP (transmission control protocol).*** Daneben gibt es das quittungslose Datagramm-Transportprotokoll ***UDP (user datagram protocol).*** Als ***Dienste*** gehören der DoD-Protokollfamilie der Filetransfer zwischen entfernten Systemen ***FTP*** (***file tranfer protocol),*** der Terminalbetrieb ***TELNET*** und ***SMTP (simply mail transfer protocol)*** für die elektronische Post an. Die DoD-Standards sind netzunabhängig. Ein weiterer korrelierter Standard ist ***SNMP (simple network management protocol).***

Das Internet

Internet ist der weltweit ***größte Netzverbund***, der jedem Teilnehmer eine nahezu grenzenlose Informations- und Kommmunikationsinfrastruktur zur Verfügung stellt. Die ***Ursprünge*** dieses Netzverbundes gehen auf ein ***militärisches Forschungsprojekt*** zurück, das in den fünfziger Jahren die Forscher beschäftigte. Zu diesem Zweck wurde die Advanced Research Projects Agency, (ARPA), gegründet. Die Zielsetzung war damals, eine ***möglichst dezentrale Kommunikationsarchitektur*** zu entwickeln als zuverlässigere Alternative zu der bis dahin genutzten leitungsorientierten Datenübertragung. Hintergrund war, dass im Krieg bei der Zerstörung einer Leitung oder eines Rechners das Kommunikationssystem unvermindert weiter funktionnieren kann. Für diese ***Kommunikationsarchitektur*** wurde Ende der 60er Jahre die ***Kommunikationsprotokollfamilie TCP/IP*** entwickelt (siehe oben).

1973 war das eigentliche ***Gründungsdatum von Internet***, als nämlich die unterschiedlichen Implementierungen von paketorientierten Übertragungsmechanismen miteinander verbunden wurden. Vier unterschiedliche paketorientierte Netzwerke - das ARPAnet, ein Satelliten-Netzwerk, ein Funk-Netzwerk und das vom XEROX-PARC (Palo Alto Research Center) entwickelte Ethernet - wurden über ein Internet mittels TCP/IP miteinander verbunden.
Die Anzahl der ***angeschlossenen Hosts*** ist von einigen Hundert zu Beginn der 80er Jahre auf ca. 100’000 im 1993 und ***über 40 Millionen im 1997 explodiert***. Das Internet hat Knoten zu allen bekannten öffentlichen, kommerziellen und forschungsorientierten Netzen wie BITNET, CompuServe, USENET, EUnet, CSNET, WIN usw. Das Internet hat sich in den letzten Jahren vom reinen Wissenschaftsnetz gewandelt und wird auch kommerziell genutzt. Es wird sich wie kein anderer Netzverbund zu einem ***Super-Information-Highway*** entwickeln mit einer Vielzahl von Diensten bis hin zu multimedialen Anwendungen. (Dies erst recht, seitdem der US-Vicepresident Al Gore seit 1994 kräftig dafür Werbung machte!) Neben ***Mail (SMTP)*** und ***Filetransfer (FTP)*** wird heute vor allem der Suchdienst ***WWW (World Wide Web)*** mit dem ***HTTP (hyper text transfer protocol)*** sehr stark benutzt.

TCP/IP - Protokollfamilie

Die TCP/IP-Protokolle wurden, wie erwähnt, von der Defence Advanced Research Project Agency (DARPA) des US-Verteidigunsministeriums (Department of Defense, DoD) mit Unterstützung des National Bureau of Standards (NBS) entwickelt. Sie beinhalten eine Reihe von ***Protokollspezifikationen*** und sogenannte '***requests for comments'*** (RFC; siehe unten).
Die wichtigsten Protokolle aus der TCP/IP-Familie sind:

- Internet Protocol (IP)
- Transmission Control Program (TCP)
- User Datagram Protocol (UDP)
- Internet Control Message Protocol (ICMP)
- File Transfer Protocol (FTP)
- Terminalprogram (Telnet)
- Simply Mail Transfer Protocol (SMTP)
- Simply Network Management Protocol (SNMP)
- Hyper Text Transfer Protocol (HTTP)

Die Entwicklung der Protokolle wurde wesentlich durch die Erfahrungen im ARPAnet beeinflußt. Das amerikanische Verteidigungsministerium verwaltet die Spezifikationen der TCP/IP-Protokollfamilie.

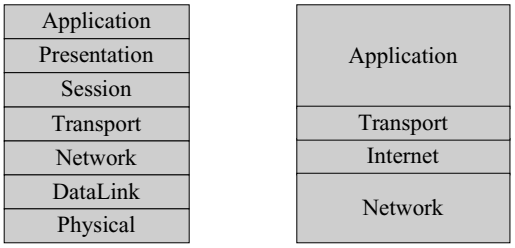
Request for Comment - RFC

Spezifikationen, Vorschläge, Ideen und Richtlinien, die das Internet betreffen, werden in Form von sogenannten RFCs (request for comments) veröffentlicht. Seit 1989 ist die ***Struktur und der Aufbau*** von RFCs durch die ***RFC 1111*** geregelt. Ein sogenannter RFC-Editor koordiniert als Mitglied des IAB die Veröffentlichung der jeweiligen RFCs sowie ihre Verteilung im Netz. Zur Zeit ist der Administrator Jan Postel (Postel@ISI.EDU). Sämtliche RFCs können im Internet angesehen werden. Im World Wide Web unter: ***http://ds.internic.net***! Die jeweils aktuellsten FTP-Archive sowie eine Anleitung zur Anforderung von RFCs via E-Mail erhält man unter 'rfc-info@ISI.EDU' mit dem Inhalt: help.
Die ***IAB (internet advisory board)*** ist eine Interessengruppe, die die Grundsätze, Leitlinien und Standards für TCP/IP und das dazugehörige Internet bestimmt hat. Das IAB wurde 1989 reorganisiert, wobei das technische Personal auf Ingenieurs- und Forschungsuntergruppen verteilt wurde. Diese Gruppen sind unter den Namen ***IETF (internet engineering task force)*** und ***IRTF (internet research task force)*** bekannt.

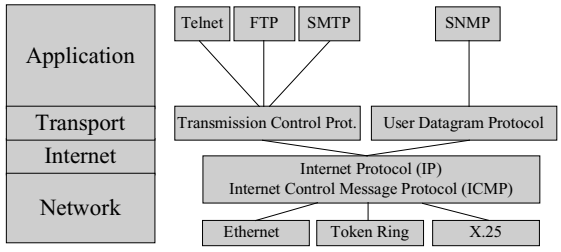
TCP/IP und das OSI-Modell

Bekanntlich dient das OSI-Kommunikationsmodell als Erklärung für Realisierungen von Netzprotokollen, die in Schichten beschrieben werden, von denen jede eine bestimmte Funktion erfüllt, aber auch die Funktion der darunterliegenden Schicht verwendet. Der Aufbau gewährleistet eine Unabhängigkeit höherer Schichten von den darunterliegenden physikalischen Bedingungen.
Da ***TCP/IP*** jedoch schon viel ***älter als das ISO/OSI-Referenzmodell*** aus dem Jahre 1983 ist, passen die Internet-Schichten nicht ganz in dieses Modell. Es gibt im Aufbau einige Gemeinsamkeiten aber auch viele Unterschiede: So wird die ***TCP/IP Netzwerkschicht*** im ***ISO/OSI***-Modell zu einer ***Bitübertragungs- und Sicherungsschicht***. Zwischen den Transport- und Anwendungsschichten sind noch die beiden ***Schichten Kommunikationssteuerung (Session Layer) und Darstellung (Presentation Layer)*** angesiedelt, die ***bei TCP/IP***-Software oft schon in den Protokollen der ***Anwendungsschicht*** integriert sind.

Beziehung zwischen dem OSI-Modell und den TCP/IP-Schichten



Beziehung zwischen den TCP/IP-Schichten und der TCP/IP-Protokollfamilie



Internet Protocol Version 4 - IPv4 (RFC 791)

Die Aufgabe des internet protocol, ***IP (OSI-Schicht 3)*** besteht darin, ***Datenpakete*** von einem Sender über mehrere Netze hinweg zu einem Empfänger zu ***transportieren***. Die ***Übertragung ist paketorientiert, verbindungslos und nicht garantiert***.

Mit verbindungslos bezeichnet man einen Datenaustausch zwischen Rechnern, bei denen zwischen den jeweiligen Teilnehmern keine Verbindung geschaltet wird. Alle anfallenden Übertragungswünsche werden in Pakete gleicher Länge organisiert (Paketvermittlung). Jedes Paket sucht unabhängig seinen Weg durch das Netz zum Empfänger (sog. Datagramm-Service) und wird von diesem nicht quittiert. Eine verbindungslose Kommunikation ist dadurch gekennzeichnet, daß nur jeweils ein einziger Informationsblock mit vollständiger Quell- und Zieladresse von einem Subsystem zu einem anderen übermittelt wird.

Die ***Datenpakete*** (hier ***Datagramme genannt***) werden vom IP als voneinander unabhängige (auch bei identischen Sendern und Empfängern) Datenpakete transportiert. ***IP garantiert weder die Einhaltung einer bestimmten Reihenfolge noch eine Ablieferung beim Empfänger*** (d.h. Datagramme können z.B. wegen Netzüberlastung verloren gehen). ***Empfangsquittungen gibt es auf der IP-Schicht nicht***. Die maximale ***Länge von IP-Datenpaketen ist auf 65'535 Bytes beschränkt***. Da bestimmte Netze, ebenso wie einige Internetworking-Komponenten, diese Paketlänge nicht verarbeiten können, spezifiziert IP die Mindestpaketlänge mit 576 Bytes. Es kann also durchaus die Notwendigkeit bestehen, IP-Datenpakete in kleinere Datenpakete aufzuspalten, zu 'fragmentieren'. Das Wiederzusammensetzen nennt man 'Reassemblieren'. Jedes IP-Fragment hat selbst wieder das Format eines gewöhnlichen IP-Paketes.

Das Internet-Protokoll unterstützt das ***Segmentieren von Nachrichten*** (Aufteilen einer großen Nachricht in mehrere kleine) und ***Reassemblieren von Nachrichten*** (mehrere kleine Nachrichtenteile werden zu einer großen Nachricht zusammengesetzt). Dadurch kann eine geforderte Transportleistung der Schicht 4 an die zur Verfügung stehende Netzleistung der darunter liegenden Schichten angepaßt werden.
Da das Internet-Protokoll ein Protokoll der Vermittlungsschicht (Network Layer) ist und diese für die Wegsuche in einem Netzwerk (Routing) zuständig ist, erlaubt das Protokoll die ***Identifikation von Stationen*** (durch Verwendung von '***Internet-Adressen'***) im Netzwerkverbund. Das internet protocol wird daher auch zur Wegsuche in komplexen Netzwerken verwendet.
Ein eigenes Feld einer Protokolldateneinheit legt die Lebensdauer ('time to live', TTL) des Datentelegramms (Datagramm) fest. Mit diesem Feld wird festgelegt, wann ein Paket verworfen wird. Der Sender des Datentelegramms wird benachrichtigt, wenn eine Einheit gelöscht wird, d.h. zu lange im Netz umhergeirrt ist!

Es existieren ***verschiedene Internet-Protokolle***, die jeweils in Zusammenhang mit einer ganzen Protokollfamilie entstanden sind. Die bekanntesten davon sind das IDP (internet datagram protocol) von Xerox aus der XNS-Familie; das von der OSI standardisierte Protokoll ISO-IP, das ***IPX (internet packet exchange) von Novell*** sowie das hier behandelte ***datagrammorientierte Internet-Protokoll IP der TCP/IP-Familie***. Heute ist IP in der Version 4 (IPv4) weitverbreitet, in Zukunft wird es die Version 6 (Version 5 fehlt!) sein**.**

IP-Protokollkopf

- allgemeine Festlegung

Nicht alle Rechnerarchitekturen haben die gleiche Art, Daten zu kodieren: deshalb ist es unumgänglich, daß für den Austausch von Paketen mit Protokollfeldern zwischen diesen Systemen ein einheitliches Format für diese Felder festgelegt wird.

- Da in den TCP/IP-Protokollfeldern sowohl einzelne Bits als auch Ganzzahlen vorkommen, ist die Bit- und Byte-Reihenfolge als 'big-endian', d.h. mit dem niederwertigsten Element an der niederwertigsten Stelle, definiert worden.
- Eine Ganzzahl, die aus mehreren Bytes besteht, wird beginnend mit dem höchstwertigen Byte gesendet (siehe Abbildung unten).

Sendereihenfolge der Bytes in TCP/IP-Protokollköpfen:

0	8	16	31
1	2	3	4
5	6	...	

IP-Protokollkopf:

1 4	8 9	16 17	25 32
Version / Länge	Servicetypen	Paketlänge	
Identifikation		0 / DF / MF	Fragmentabstand
Lebenszeit TTL	Transportprot. ID	Kopfprüfsumme	
Senderadresse			
Empfängeradresse			
Optionen			Füllzeichen (PAD)

Die Felder des oben dargestellten Protokollkopfes haben folgende Bedeutung:

Versionsnummer: Dieses vier Bit breite Feld enthält die Versionsnummer des IP-Protokolls, die derzeitige Versionsnummer ist 4. (Bei IPng wird es die Nummer 6 sein!)

Länge: Länge des IP-Protokollkopfes in 32-Bit-Worten. Der kleinste IP-Protokollkopf enthält 5 Worte; da das auch der Normalfall ist, beginnen nahezu alle TCP/IP-Pakete mit $45_{(Hex)}$. Die Länge des Protokollkopfes kann sich durch Anfügen von Optionsfeldern vergrößern, daher muß für die Interpretation die genaue Länge bekannt sein.

Servicetypen: Angaben für den IP-Protokollautomaten, die Nachricht nach bestimmten Kriterien zu behandeln. In der Praxis verwendet man fast immer den Wert 0, da selten mehrere qualitativ unterschiedliche Pfade zwischen zwei Rechnern existieren.

Paketlänge: Enthält die Länge des Pakets inklusive des Protokollkopfes. Diese Angabe wird zur Feststellung der Datenlänge verwendet. Da dieses Feld 16 Bits breit ist, kann ein IP-Paket inklusive Protokollkopf nur maximal 65535 Bytes (2^{16}) lang werden. Die IP-Spezifikation legt fest, daß jeder Host in der Lage sein muß, Pakete bis 576 Bytes Länge zu empfangen und zu reassemblieren (sprich: wieder zusammenzusetzen). In der Regel können Rechner aber wesentlich größere Pakete, mindestens aber bis zur Maximalgröße des angeschlossenen Netzwerks (z.B. Ethernet-Paket) verarbeiten.

Identifikation: Eine eindeutige Identifikation, z.B. ein Zähler, der durch den absendenden Host vergeben wird. Dieses Feld wird bei der Reassemblierung von Fragmenten verwendet, um alle Teile einer Fragmentkette identifizieren zu können.

Flags: Zwei Bits namens DF ('don't fragment') und MF ('more fragments') steuern die Behandlung des Pakets im Falle der Fragmentierung. Ist das DF-Bit gesetzt, darf das IP-Paket unter keinen Umständen fragmentiert werden, auch wenn es dann nicht mehr weitertransportiert werden kann und weggeworfen werden muß. Das MF-Bit zeigt an, ob dem IP-Paket weitere Teilpakete nachfolgen. Das erste Bit dieses Feldes ist ungenutzt.

Fragmentabstand: Wenn das MF-Bit gesetzt ist, enthält der Fragmentabstand die Lage der in diesem Paket gespeicherten Teilnachricht relativ zum Beginn der Gesamtnachricht. Mit Hilfe dieser Angabe kann der empfangende Host das Originalpaket wieder richtig zusammensetzen. Weil dieses Feld aufgrund der vorangestellten Flags nur 13 Bits groß ist, muß der Abstand in Einheiten von 8 Bytes gezählt werden, nur so läßt sich die Maximallänge eines IP-Pakets ($8 * 2^{13} = 65535$) darstellen.

Lebenszeit: Der absendende Host gibt hier an, wie lange das Paket im Netz verweilen darf, bevor es weggeworfen werden muß. RFC 79l spezifiziert die in diesem Feld verwendete Einheit in Sekunden, schreibt aber vor, daß jeder Router dieses Feld auch dann mindestens um 1 verringern muß, wenn die Durchlaufzeit unter 1 Sekunde liegt. Somit ist die Lebenszeit (TTL = 'time to live') meist gleichbedeutend mit der Anzahl der Netzknoten, die von einem Paket maximal durchlaufen werden können. Wenn dieses Feld '0' enthält, muß das Paket vom verarbeitenden Rechner weggeworfen werden; somit wird verhindert, daß ein Paket endlos im Netz zirkuliert. Der Absender des Pakets erhält in diesem Fall eine ICMP-Nachricht über den Vorgang. UNIX-Rechner setzen dieses Feld in der Regel auf einen Wert zwischen 15 und 30, wobei bei LAN-Anwendungen das Feld typisch auf 'FF' gesetzt wird, in der Annahme, im LAN seien Lebenszeit / Sicherheit kein Problem.

Transportprotokoll: Enthält die Identifikation des Transportprotokolls, dem dieses Paket zugestellt werden muß. Für TCP ist z.B. die Nummer 6, für UDP 17 und für ICMP 1 festgelegt. Diese Nummern werden von NIC (Network Information Center; NIC.DDN.MIL) in regelmäßig erscheinenden RFCs festgelegt, derzeit existieren ca. 50 offizielle höhere Protokolle.

Kopfprüfsumme: Dieses Feld enthält die Prüfsumme der Felder im Protokollkopf. Mit einer entsprechenden Gegenprüfung läßt sich verhindern, daß Netzknoten oder Hosts mit verfälschten Daten arbeiten. Die Nutzdaten im IP-Paket werden aus Gründen der Effektivität nicht geprüft, diese Prüfung findet beim Empfänger innerhalb des Transportprotokolls statt. Da das IP-Paket durch die Verringerung des Lebenszeitfelds in jedem Netzknoten verändert wird, ist eine effiziente Bildung der Prüfsumme wichtig.

Sender- und Empfängeradresse: Hier werden die 32-Bit langen Internet-Adressen - deren Aufbau auf den nächsten Seiten näher erklärt wird - eingetragen.

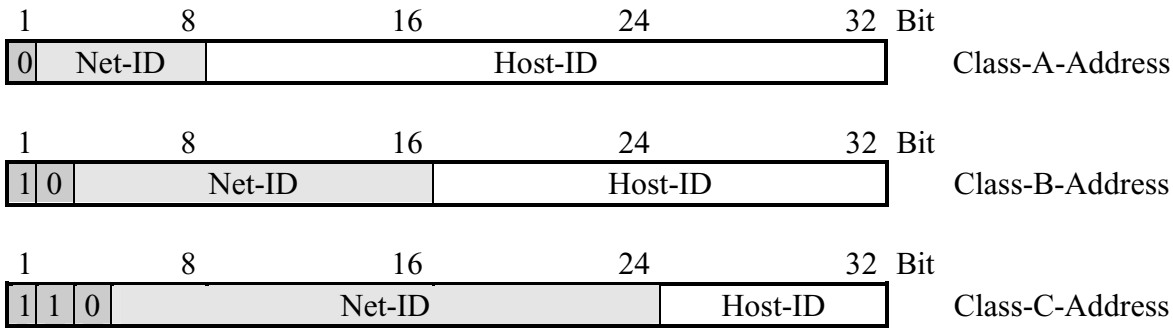
Optionen und Füllzeichen: Für Spezialaufgaben (Netzwerkmanagement, Sicherheit) wird der IP-Kopf um Optionen erweitert. Damit die Zahl der 8-Bit-Worte im IP-Kopf immer ein Vielfaches von 4 ist - wir erinnern uns an das Kopflängenfeld - wird er nötigenfalls durch Füllzeichen aufgefüllt.

Internet-Adressierung (IP-Adresse)

IP-Adressen sind die Abstraktion der physikalischen Hardwareadressen, wie ein Internet die Abstraktion eines physikalischen Netzwerks ist. So besteht die ***Adresse eines Hosts*** jeweils aus einem ***Netzwerkteil*** und einem ***Hostteil.*** Durch diese Aufteilung kann das Routing effektiver gemacht werden.

Eine ***Internet-Adresse*** besteht aus ***32 Bits*** (4 Bytes zu 8 Bit). Im allgemeinen werden ***drei Adreßklassen*** unterstützt (Der Vollständigkeit halber seien hier noch die Class-D- und Class-E-Adressen erwähnt. Die Class-D-Adresse dient für spezielle Multicasts, welche in der Praxis jedoch sehr wenig verwendet werden. Die Class-E-Adresse dient Forschungszwecken):

Internet-Adressformate:



- Class-A-Adresse:

7 Bits werden hier für die Netzwerknummer und 24 Bits für die lokale Nummer (Host-Nummer) reserviert (erstes Bit zwingend '0'). Class-A-Adressen werden verwendet für sehr grosse ***Netzwerke mit bis zu 16 Millionen Rechner***, es sind jedoch nur ***128 Class-A-Netzwerke*** möglich (***Netzwerkskennung (netid): 0-127***)!

Format: 0NNN'NNNN'HHHH'HHHH'HHHH'HHHH'HHHH'HHHH$_2$
oder Dezimal: N.H.H.H$_{10}$
(N = Netzwerkadresse, H = Hostadresse)
Beispiel: 0000'1100'0000'0000'0000'0011'1111'0001$_2$
0C'00'03'F1$_H$
12.0.3.241 (Netz-ID 12, Host-ID 0.3.241; übliche Darstellung)

- Class-B-Adresse:

Hier werden 14 Bits für die Netzwerknummer und 16 Bits für die lokale Nummer reserviert (ersten beiden Bits zwingend '10'). Somit können ***16'384 Class-B-Netzwerke*** mit jeweils ***65'536 Knoten*** kreiert werden (***Netzwerkskennung (netid): 128.0 bis 191.255***).

Format: 10NN'NNNN'NNNN'NNNN'HHHH'HHHH'HHHH'HHHH$_2$
N.N.H.H$_{10}$
Beispiel: 1000'1100'0000'0000'0000'0011'1111'0001$_2$
8C'00'03'F1$_H$
140.0.3.241 (Netz-ID 140.0, Host-ID 3.241)

- Class-C-Adresse:

Bei 21 Bits für die Netzwerknummer und nur 8 Bits für lokale Adressen (ersten drei Bits zwingend '110') können ***ca. 2 Millionen Netze*** mit jeweils ***maximal 256 Knoten*** definiert werden (***Netzwerkskennung (netid): 192.0.0 bis 223.255.255***).

Format:	110N'NNNN'NNNN'NNNN'NNNN'NNNN'HHHH'HHHH$_2$
	N.N.N.H$_{10}$
Beispiel:	1100'1100'0000'0000'0000'0011'1111'0001$_2$
	CC'00'03'F1$_H$
	204.0.3.241 (Netz-ID 204.0.3, Host-ID 241)

Eine Internet-Adresse identifiziert einen Rechner im Netzwerk. Rechner mit mehreren physikalischen Anschlüssen (Multi Homing Host) haben mehrere Adressen. Es gibt ***reservierte Adressen*** für spezielle Aufgaben - eine ***Adresse mit der Nummer '0' identifiziert das lokale Netzwerk.*** Die Klasse-A-Adresse ***127*** ist für ein sogenanntes ***Loopback*** reserviert. Ein Paket mit der Zieladresse 127.0.0.1 wird deshalb, ohne jemals auf das Netzwerk zu gelangen, unmittelbar wieder beim Sender eintreffen. Loopback-Pakete können zum Funktionstest von Netzwerksoftware verwendet werden. Adressen, bei denen ***jedes Bit auf '1'*** (FF'FF'FF'FF) gesetzt ist, sind reserviert für ***Broadcast-Nachrichten***. So lautet beispielsweise die Broadcast-Adresse für das Netzwerk 72 somit 72.255.255.255 (acht binäre Einsen, dezimal ausgedrückt, ergeben 255!).
Nachteilig wirkt sich bei diesem Adressierungs-Schema die Änderung des Standortes eines Rechners aus, da sich von einem Netzwerk zu einem anderen die Internet-Adresse des Rechners ändert (siehe dazu DHCP weiter unten).

Zwischen 1995 und 1997 erfuhr ***TCP/IP*** eine neuerliche ***Überarbeitung*** durch die IETF. Die TCP/IP V6 (IPNG - IP next generation) Definition wurde 1997 abgeschlossen. Produkte sind heute verfügbar. Da jedoch seitens der Netzwerkverantwortlichen einiges an Aufwand bei einer Migrierung geleistet werden muss werden wohl noch längere Zeit die neue Version und die alte Version koexistieren (siehe dazu IPv6 weiter hinten).

Subnet-Adressmasken

Große Unternehmen betreiben Netzwerke, die aus mehreren tausend Benutzern und vielen hundert Teilnetzwerken (Subnetworks) bestehen. Da sie jedoch nicht auf den Vorteil einer einzigen Internetadresse verzichten wollen, wird ein Teil der verfügbaren Benutzeridentifikation dazu verwendet, die Internetadresse in mehrere Subnetzwerke zu strukturieren. Ein Netzwerk mit der Klasse-B-Adresse 128.69 könnte so beispielsweise in die Subnetzwerke A und B aufgeteilt werden, so daß Subnetz A nur den Adressenbereich 128.69.1.1 bis 128.69.1.254 benutzt, und Subnetz B den Bereich 128.69.2.1 bis 128.69.2.254. Die Verbindungskomponente zwischen beiden Netzwerken, ein Router, kann nun mittels einer sogenannten Subnet-Adressmaske leicht feststellen, ob ein Datenpaket des einen Netzwerkes ein lokales Ziel besitzt, in das zweite Subnetzwerk vermittelt (geroutet) oder überhaupt in ein anderes Internet-Netzwerk übertragen werden soll. Die Subnet-Adressmaske für unser obiges Beispiel wäre 255.255.255.0. Sie ist also nichts anderes als eine Bitmaske, die Nullen für den Adressenbereich enthält, der für den lokalen Subnetzbereich vorgesehen ist, und Einsen für alle anderen Bits. In unserem Beispiel interessieren den Router also nur die ersten drei Bytes, da anhand dieser bereits festgestellt werden kann, ob ein bestimmtes Paket weitervermittelt werden muß oder nicht.
Falsch gesetzte Subnet-Adressmasken sind häufig die Ursache von Kommunikationsproblemen. Würde in unserem Fall die Subnet-Adressmaske irrtümlich auf 255.255.0.0 gesetzt, würde dies bedeuten, daß alle Datenpakete mit den Zieladressen 128.69.xxx.xxx nicht vermittelt werden. (Der Router überprüft lediglich die ersten beiden Bytes, und diese zeigen ihm an, daß sich das Paket bereits im richtigen Netzwerk befindet!). Jegliche Kommunikation mit dem Nachbarnetzwerk wäre unterbunden!

Zu erwähnen ist in diesem Zusammenhang der ***Default Gateway***. Der Default Gateway ist ein Router, dessen IP-Adresse bei der Konfiguration eines Rechners jeweils angegeben werden muss. Denn wenn die Zieladresse im lokalen Netzwerk nicht gefunden wird, wird die Nachricht vom Default Gateway bearbeitet , der dann den richtigen Weg zum Zielort zu finden hat.

Subnet-Maske für B-Adresse

B-Adresse 140.0.3.241	1100’1100’0000’0000’0000’0011’1111’0001
B-Adr-Maske 255.255.0.0	1111’1111’1111’1111’0000’0000’0000’0000
B-Class-Netzadr. 140.0	1100’1100’0000’0000

Durch die B-Adress-Maske wurde die Node-Id ‘3.241’ vom Rest der IP-Adresse abgetrennt.

Subnet-Maske für C-Adresse

C-Adresse 204.0.3.241	1100’1100’0000’0000’0000’0011’1111’0001
C-Adr-Maske 255.255.255.0	1111’1111’1111’1111’1111’1111’0000’0000
C-Class-Netzadr. 204.0.3	1100’1100’0000’0000’0000’0011

Durch die C-Adress-Maske wurde die Node-Id ‘241’ vom Rest der IP-Adresse abgetrennt.

Internet-Protokoll Version 6 - IPv6 (RFC 1752)

IPv6 - Warum ein neues Protokoll?

Das explosionsartige Wachstum des Internets, die Adressvergabe und der Anbindungswunsch vieler Unternehmen bilden einen der heutigen Problembereiche. Mit der 32-Bit Adressbreite im IPv4 kann der wachsende Bedarf nach Internet-konformen Adressen nicht mehr lange gedeckt werden. Bereits jetzt ist die Adresszuweisung umständlich und ineffizient. Eines der Hauptprobleme ergibt sich durch die Nutzung nicht-konformer Adressen in den TCP/IP Netzwerken, die noch nicht an das Intemet angeschlossen sind. Nach Schätzungen benutzen 60% der Unternehmen solche Adressen als Hinterlassenschaft aus den Zeiten, als das Intemet noch eine unbedeutende Rolle spielte.
Andere Schwächen sind die Sicherheit und der in Spitzenbelastungen eher geringe Durchsatz. Ein Schutz vor unerlaubten Eindringen in Netzwerke, illegaler Datenverwendung oder sogar Datenzerstörung ist für Untemehmen unabdingbar. Die zunehmenden Online-Dienste und Transaktionen mit direkter System- und Netzwerkanbindung erfordern die eindeutige Authentifizierung und Autorisierung der Benutzer und die Verschlüsselung der Daten.
Die Wünsche nach Echtzeit-Anwendungen wie beispielsweise Videokonferenzen über das Netzwerk können mit IPv4 kaum erfüllt werden. Das neue Protokoll IPv6 bietet für all diese Probleme eine Reihe von Lösungen an.

Die IPv6 Historie

Mit der stark wachsenden Anzahl der Teilnehmer im Internet und deren Anforderungen wurde bereits 1991 klar, daß eine Lösung gefunden werden mußte. Sicherheit, Leistung, Vereinfachung der Administration und des Routings bei gleichzeitiger Verbreiterung des Adressraums waren die Schwerpunkte.
Die IETF (Internet Engineering Task Force) führte eine Hochrechnung zur Ermittlung des Wachstums des Internets durch. Auf dieser Basis begann man im Juli ’96 im Rahmen der IP next generation (IPng), Designvorschläge für das neue Protokoll zu erarbeiten.
Zwei Jahre später wurden diese Empfehlungen für das 128-Bit IPv6 genehmigt. Ein Jahr danach wurden die Spezifikationen verabschiedet. Sie sind im RFC (Request for Comment) Nummer 1752 ‘The Recommendation for the IP Next Generation Protocol’ dokumentiert. In den RFCs werden auch alle Anwenderanforderungen gesammelt. Damit ist ein einseitiger Einfluß auf Standards weitgehend ausgeschlossen. Alle führenden Netzwerkanbieter haben dem neuen IPv6 Standard zugestimmt. Damit sind alle Zweifel ausgeräumt: IPv6 ist der Standard der Zukunft.

Was ist neu in IPv6?

Neu: Struktur der Adressen

Mit IPv6 wird eine neue und sehr effiziente Adressierung eingeführt. Die Adressen sind skalierbar und ermöglichen eine erhebliche Verbesserung des Datendurchsatzes.
Den Anwendern sind zumeist die logischen Adressen bekannt. Sie werden über Domain Name Server (DNS) verwaltet, z. B. boesiger@bkw.ch. Diese einprägsamen Namen werden mit den echten IP-Adressen verknüpft, die sich aus einer Reihe von Ziffern und Punkten zusammensetzen. Bislang standen mit IPv4 nur 32 Bit zur Abbildung einer Geräteadresse zur Verfügung (entspricht 4’294’967’296 möglichen Adressen). Mit IPv6 sind das 128 Bit und damit 3,4 x 10^{38} Adressen (oder anders ausgedrückt: 340’282’366’920’938’000’000’000’000’000’000’000’000!). Genug, um jeden m^2 der Erdoberfläche mit 665’570’793’348’866’943’898’599 Adressen zu versorgen. Das erscheint Ihnen ein wenig zuviel? Dieser gewaltige Adressraum bietet jedoch entscheidende Vorteile:

- *Effizientes Routing:* Die Adressstruktur läßt die Abbildung tief gestaffelter hierarchischer Adressen zu. Damit wird das Routing sehr viel einfacher und schneller, weil die Routing-Tabellen reduziert werden.

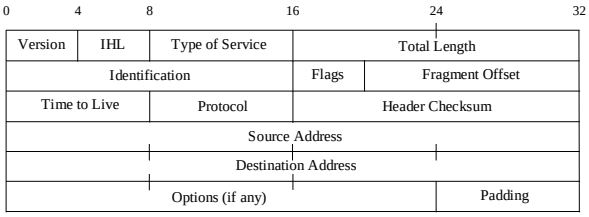
- *Selbstkonfiguration:* Die Selbstkonfiguration eines Netzwerks setzt immer eine bestimmte Bandbreite noch freier Adressen voraus, die in der Regel größer ist als die Anzahl der Geräte. Dieses Prinzip ist in IPv6 realisiert.

- *Problemlose Adressvergabe*: Durch die Vielzahl und Struktur der Adressen gegeben.
Adresskonventionen / -Notationen:

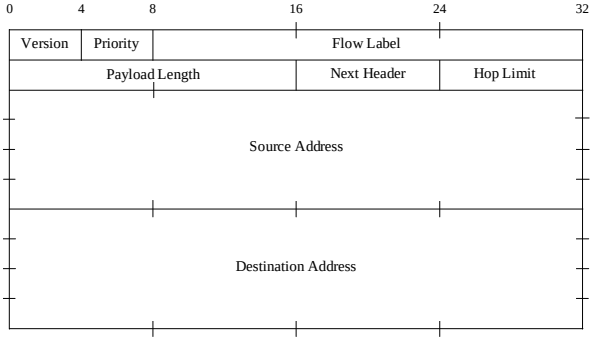
IPv4: 4 Dezimal-Nummern, getrennt durch einen Punkt.
Beispiel: ***192.232.11.53***
IPv6: 8 Hexadezimal-Nummern, getrennt durch einen Doppelpunkt.
Beispiel: ***FEC:BA9:0:0:0:0:7654:310***
Kurznotation für '0'-Folge mit 'Doppelpunkt-Paar'
Beispiel: FEC:BA9::7654:310

- ***Migration:*** IPv4 Netzwerke können parallel zu IPv6 existieren. Ein Teil des IPv6 Adressraums ist für IPv4 Adressen reserviert. Es gibt darüber hinaus eine Vereinbarung über die Generierung von IPv4 Adressen als IPv6 kompatible Adressen. Anwender kommunizieren also transparent miteinander.
Eine Umstellung von IPv4 auf IPv6 muss sorgfälltig geplant werden. Es wird keinen definierten Tag geben, an dem ein Schalter umgelegt und damit alle IPv4 Netzwerke auf IPv6 umgestellt werden. Bereits bei der Definition von IPv6 wurde eine schrittweise Umstellung berücksichtigt. IPv4 und IPv6 können gleichzeitig benutzt werden.
Man hat die Möglichkeit zur reibungslosen Kommunikation mit allen IPv4 Geräten und beginnt gleichzeitig mit künftigen Netzwerkanwendungen. Vorteile wie die erweiterte Sicherheit (die Abwicklung von Geschäften über das Netzwerk ist möglich), Selbstkonfiguration (Kostensenkung für die Netzwerkadministration) und höherer Datendurchsatz (neue Echtzeit-Anwendungsbereiche) können genutzt werden. Und das mit dem bestehenden Netzwerk.

IPv4 - Header



IPv6 - Header



Der Vergleich der Header macht die Vorteile von IPv6 deutlich: vierfacherAdressraum, nur sechs Felder (mehr Geschwindigheit und Effizienz) sowie zwei Adressen und keine weiteren Optionen. Das bedeutet eine feste Headerlänge und ein erheblich effizienteres Routing.

Neu: Selbstkonfiguration
Ein TCP/IP-Rechner kann bei IPv6 automatisch mit einer Adresse konfiguriert werden. Wenn ein Router oder ein Adress-Server mit IPv6 Implementierung vorhanden ist, findet der Rechner mit deren Hilfe eine nur für das jeweilige Subnetz gültige 'local link Address'. Danach wird diese Adresse durch ein 'unique token' zu einer eindeutigen Adresse vervollständigt. Mit dieser Adresse wird nun im Netz nach einem Duplikat gesucht. Ist keines vorhanden, wird die

Adresse als neue, gültige IP Adresse im Router oder Server eingetragen. Dieser Vorgang geschieht ohne manuelles Eingreifen. Ist bei größeren Netzen eine Zuweisung von Adressen erforderlich, wird der Rechner einem Address-Server wie z. B. einem DHCP-Server zugeordnet, der dann die Adresse vergibt. Nach diesem Prinzip funktioniert auch die Selbstkonfiguration in älteren IP Netzwerken mit nicht-konformen Adressen. Damit können also alle IP Netzwerke miteinander verbunden werden. Vorteile der Selbstkonfiguration: schnelle, unproblematische Installation kleinerer Netze, erhebliche Einsparungen bei der Administration größerer IP Netzwerke.

Neu: Multimedia und Echtzeit Anwendungen
IPv6 erschließt neue Anwendungsbereiche: Multimedia, Video- und Audiokonferenzen, die unter IPv4 eher stockend verlaufen. Für diese Anwendungen ist eine hohe Uebertragungsrate erforderlich. Die konventionelle TCP Kontrolle ist zu langsam und hierfür nicht geeignet. Durch unterschiedliche Geschwindigkeiten der Netzwerkanbindungen (LAN, Modem, ISDN,...) entstehen zusätzliche Schwierigkeiten.
Das Design des neuen IPv6 Header bietet die elegante Lösung. Ein 'Flow Label' ermöglicht den Austausch von Paketen zwischen zwei Adressen mit besonderer Behandlung: dem 'Resource Reservation Protocol' (RSVP). Damit wird die erfolgreiche Anlieferung sichergestellt. Die Geschwindigkeit wird gesteigert, weil der 'Flow Label' Bestandteil des Main Headers ist, eine feste Länge besitzt und damit ein erheblich effizienteres Routing ermöglicht. Ergänzt wird dieser Mechanismus durch das Feld 'Drop Priority', mit dessen Hilfe nach Prioritäten sortiert werden kann; zeitlich kritische Daten werden bevorzugt geliefert

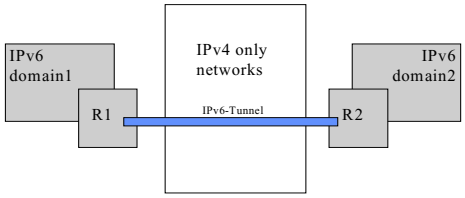
Neu: Sicherheit
Als komplexes Thema umfaßt Netzwerksicherheit eine Vielzahl von Technologien, mit einer Bandbreite vom Stack bis zur Anwendung, vom Zugang bis zur Datenintegrität. Wir beschäftigen uns hier nur mit der durch IPv6 erweiterten Sicherheit in IP Netzen.
Sichere Kommunikation ist mit IPv6 möglich. Ueber die Authentication weiß der Empfänger genau, woher eine Nachricht kommt. Die Verschlüsselung stellt sicher, daß nur der Empfänger sie lesen kann, für den sie bestimmt ist. IP-SEC bezeichnet die durch IPv6 auf der Kernel-Ebene implementierten Sicherheitsfunktionen. Optionale Header bieten Authentication durch einen 'Authentication Header' (AH) und Verschlüsselung durch die 'Encapsulating Security Payload' (ESP). Der Authentication Header schützt Netzwerke vor den Risiken durch 'source-routing' und 'host masquerading attacks', während 'Encapsulating Security Payload' die Vertraulichkeit der Informationen sichert.
Digitale Unterschriften sind die Basis für die IP-SEC-Sicherheit. In der Spezifikation für IP-SEC sind die notwendigen Parameter für die Authentifizierungs- und Verschlüsselungs-Algorithmen und ein Schlüssel-Distributionssystem definiert. Die Voreinstellung ist die Authentifizierung über MD5 und die Verschlüsselung über 'Date Encryption Standard' (DES). Die Nutzung anderer Mechanismen ist nicht vorgesehen. Für die Anwender außerhalb der USA ist das besonders wichtig, da für die standardmäßige 56-Bit-Verschlüsselung eine Genehmigung zum Export erforderlich ist. Es wird erwartet, daß diese Genehmigungspflicht bald aufgehoben wird, da sie einer vollständigen Implementation der IPv6 Sicherheit im Wege steht. IP-SEC auf der Kernel-Ebene ist für alle Anwendungen verfügbar und transparent. Der Umstieg auf IPv6 ist erheblich einfacher und kostengünstiger als die nachträgliche Implementierung dieser oder ähnlicher Sicherheitsfunktionalität in IPv4.

Die Migration auf IPv6

Der Umstieg auf IPv6 wird und muß nicht über Nacht geschehen. Die installierte Basis mit IPv4 und das Intemet mit den Millionen von Knoten machen das unmöglich. Statt dessen ist die sanfte und stufenweise Migration möglich und vorgesehen. Eine sehr elegante Methode ist die sogenannte 'Dual Stack' Strategie. Damit kann ein auf IPv6 aufgerüsteter Host sowohl über

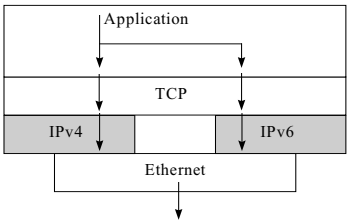
IPv4 als auch IPv6 kommunizieren und wird daher zwei Adressen haben, eine für jedes Protokoll. Eine andere Möglichkeit ist das ‘Tunneling’. Damit können ‘Inseln’ von IPv6 Hosts, ob einzelne PCs oder ganze Unternehmensnetzwerke, mit anderen IPv6-Inseln über IPv4 Netzwerke und Router durch IP-Tunneling kommunizieren. Die Transitionsstandards für IPv6 sind im RFC 1933 'Simple Internet Transition Mechanism' dokumentiert. Danach können Hosts und Router unabhängig voneinander den Umstieg auf IPv6 vollziehen. Die einzige Voraussetzung ist ein IPv6 DNS Server, der sowohl IPv4 als auch IPv6 Adressen unterstützt. Man kann die Migration auch über den Einsatz von Gateways beginnen, die sofort die transparente Kommunikation zwischen IPv4 und IPv6 ermöglichen.

IPv4 - Tunneling



Mit IPv6-fähiger Software und einem IPv6 Router kommunizieren Benutzer über einen Tunnel durch das IPv4 Netzwerk. Die Datenpakete werden in die IPv4 Pakete ‘eingepackt’ und zwischen den beiden Routern ausgetauscht.

Dual-Stack-Implementierung



Die Koexistenz zwischen IPv4 und IPv6 wird über die Dual-Stack-Implementierung hergestelt. Dadurch erhält der Node zwar zwei Adressen und verwaltet auch ein doppeltes Set, aber die Selbstkonfguration ist der entscheidende Vorteil, der diesen zusätzlichen Aufwand ausgleicht.

Dynamic Host Configuration Protocol - DHCP (RFC 1533)

Das DHCP wurde entwickelt, damit TCP/IP-Konfigurationen zentral vorgenommen werden können und daß die ***IP-Adressen automatisch*** den jeweilig angeschlossenen Computern ***zugeteilt*** werden können. Um dies realisieren zu können, braucht es zwei Arten von DHCP-Rechnern: den ***DHCP-Server*** und den ***DHCP-Client***. Ein DHCP-Server besitzt einen Pool von IP-Adressen über die er frei verfügen kann. Hat nun ein neu ans IP-Netwerk anzuschliessender Rechner einen DHCP-Client geladen, so fragt dieser beim Hochfahren den DHCP-Server um eine IP-Adresse nach. Sofern noch vorhanden, stellt ihm der DHCP-Server aus seinem Pool eine ungenutzte IP-Adresse zur Verfügung.
DHCP stellt insbesondere bei Notebooks einen grossen Fortschritt dar. Ohne DHCP muss bei einem mobilen Gerät jedes mal von Hand eine andere IP-Adresse eingestellt werden, wenn der Rechner in einem anderen Segment ans Netzwerk angeschlossen wird!
Damit nicht ein unnötiger zusätzlicher Netzverkehr erzeugt wird, kann man verschiedene Möglichkeiten bei der Konfiguration vorsehen. Die 'Lebenszeit' der Client-IP-Adresse kann gewählt werden (von 'bis zum Ausschalten des Clients' bis 'bleibt fest bestehen'). Aus dem Server-Pool können Adressen ausgeklammert werden, denn bspw. ein Server sollte doch eine feste, reservierte Adresse haben!
DHCP ist ein relativ neues Protokoll, das erst jetzt grössere Verbreitung findet. Microsoft beispielsweise bietet den DHCP-Server auf dem NT-Server ab Version 4 an.

Address Resolution Protocol - ARP (RFC 826)

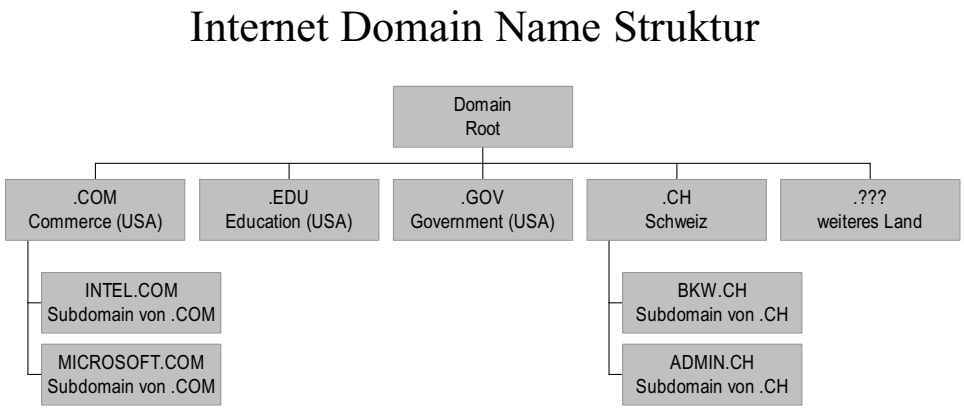
Die gebräuchlichste Methode, ***Internet-Adressen in Ethernet-Adressen umzuwandeln***, ist die ARP-Methode, die das address resolution protocol verwendet. Vor der Übertragung von Daten über das Ethernet fragt IP bei ARP nach der Ethernet-Adresse der zugehörigen Ziel-Internet-Adresse an. ARP vergleicht seine Adreßtabellen (auch ARP-Tabellen oder Internet nach Ethernet-Translation-Tabellen genannt) mit der Anfrage. Hat ARP keinen Eintrag in seiner Tabelle, so wird über eine Anfrage an alle Netzknoten (broadcast) die Ethernet-Adresse der zugehörigen Internet-Adresse erfragt. Nur Netzknoten mit einem Eintrag zu dieser Internet-Adresse antworten auf die Anfrage. Die Antwort auf den ARP-Broadcast wird in der lokalen ARP-Adreßtabelle gespeichert (nicht zu verwechseln mit der 'Host-Table').
Hinweis: Bei NT/Unix kann mit dem Command 'arp -a' der Cacheinhalt eingesehen werden.

Reverse Address Resolution Protocol - RARP (RFC 903)

Das Reverse-address-resolution-Protokoll wird benutzt, um einer diskless Workstation eine ***IP-Adresse zuordnen*** zu können. Das RARP arbeitet auf der Vermittlungsschicht (Network Layer) und ist ***aus dem Address-Resolution-Protokoll abgeleitet***. Da aber ohne IP-Adresse keine netzwerkweite Kommunikation möglich ist, muß sich die Station von einem anderen Rechner (RARP-Server) eine IP-Adresse holen.

Domain Name Service - DNS (RFC 1034)

DNS ist ein online verteiltes ***Datenbanksystem***, das in der Lage ist, ***von Menschen lesbare Maschinennamen in IP-Adressen aufzulisten*** (bspw. BOR.AI.BKW nach 160.80.40.10). Durch das ganze zusammengeschaltete Internet bieten DNS-Server einen hierarchisch geordneten Namensraum, um Firmen die Möglichkeit zu geben, Maschinennamen und Adressen selbst zu bestimmen. DNS unterstützt auch verschiedene Verzeichnislisten zwischen der elektronischen Post (E-Mail) und IP-Adressen.
Wie bei der numerischen Schreibweise gliedert sich im Domain-Namensystem der Adressname in einen ***Benutzerteil (user-id)*** und einen ***Netzwerkteil***, der als Domain-Name bezeichnet wird. Die einer anderen Domain hierarchisch untergeordnete Domain wird als SubDomain bezeichnet. Der Domain-Name besteht demnach aus einer sogenannten Top-Level-Domain (der am weitesten rechts befindliche Name) sowie aus Sub-Domains. So ist der zugehörige Internetadressname für die Adresse 160.50.40.210 beispielsweise 'bkw.ch'. (Obwohl auch beim Adreßnamensystem die einzelnen Adreßteile durch einen Punkt unterteilt sind, korrespondieren diese Teile nicht mit den ebenfalls durch Punkte getrennten Bytes einer numerischen Internet-Adresse!)
Jede Benutzeridentifikation innerhalb eines Netzwerkes muß dabei einzigartig sein. Dasselbe gilt für den Domainnamen innerhalb des jeweils übergeordneten Netzwerkes. Es darf also innerhalb des Netzwerkes 'bkw.ch' nur einen einzigen Node 'bor' geben, genauso wie es innerhalb von 'ch' nur ein einziges Netzwerk mit dem Namen 'bkw' geben darf.
Domainnamen dürfen höchstens zwölf Buchstaben lang sein und sind in der Regel identisch mit dem Namen der Organisation oder des Unternehmens (bkw.ch). Im allgemeinen werden Domainnamen klein geschrieben. Es werden jedoch auch Grossbuchstaben oder eine Mischung von Gross und Klein erkannt.
Da Domainnamen eindeutig sein müssen, braucht es eine Clearing-Stelle. In der Schweiz wurde diese Aufgabe vom BAKOM an SWITCH übertragen (gilt für alle xxx.ch Adressen). Die Registrierung kann über Internet erfolgen: ***http://www.nic.ch***.



Host Tabellen

Steht in einem Netzwerk kein Namenserver zur Verfügung, so muss entweder der mühsame Weg über die explizite lP-Adresse gegangen werden, oder es wird eine lokale Umsetzungstabelle erstellt. Diese nennt man Host-Table (unter Unix: /etc/hosts; bei NT: ...\net\lmhosts). Ein Eintrag in der Host-Tabelle besteht aus einer Internet-Adresse im Dezimal-, Hexadezimal oder Oktalformat und einem offiziellen Namen (siehe Beispiel unten). Zusätzlich können nach dem offiziellen Namen ein oder mehrere Aliasnamen vergeben werden. Der Netzverwalter trägt hier (manuell) neue Adressen, Namen und Aliasnamen ein, wenn weitere Rechner ans Netz angeschlossen werden.

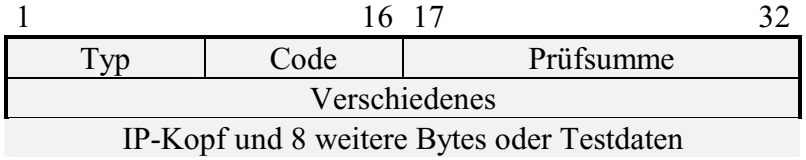
Beispiel einer Host-Tabelle:

```
. local network host addresses
127.0.0.1              local   localhost
192.10.100.301         ch1     CH1
192.10.100.302         ch2     CH2
0xC0.0x09.0x96.0x01    ch7             . hexadecimal
0300.011.0226.02       tsbe4           . octal
```

Internet Control Message Protocol - ICMP (RFC 792)

ICMP ist ein Protokoll zur ***Übertragung von Information, Diagnosen und Fehlermeldungen*** zwischen IP-Netzknoten auf Schicht 3. Wird von IP jedoch wie ein Protokoll höherer Schichten behandelt. ICMP ist ein ***integraler Bestandteil des IP-Protokolls*** von TCP/IP. Besonders Gateways und Hosts benutzen ICMP, um Berichte über Probleme mit Datagrammen (UDP-Format siehe weiter hinten) zur Originalquelle zurückzuschicken. ICMP bietet außerdem die Möglichkeit einer Echo-Anforderung, um feststellen zu können, ob ein Bestimmungsort erreichbar ist und antwortet (das bekannte 'Ping' Kommando, siehe weiter unten).

ICMP - Protokollkopf:



Typenfeld: Spezifiziert die Art der ICMP-Nachricht. Eine Auswahl davon:

Typenfeld	Funktion	Typenfeld	Funktion
0	Echo Reply	11	Time Exceeded for a Datagramm
3	Destination Unreachable	12	Parameter Problem on a Datagram
5	Redirect	13	Timestamp Request
8	Echo Request	14	Timestamp Reply

Code: Ein weiterer Code, der Unterfunktionen innerhalb eines Typs darstellt (wird in dieser Übersicht nicht weiter erläutert).

Prüfsummenfeld: Internet-Prüfsumme der gesamten ICMP-Nachricht.

Verschiedenes: 32 Bit langes Feld, das Informationen für verschiedene Zwecke (Sequenznummer, Internet-Adressen usw.) aufnehmen kann.

IP-Protokollkopf: Enthält das auslösende IP-Datagramm und die ersten 8 Byte der darin transportierten Nachricht. Wurde die ICMP-Nachricht durch eine Nachricht von TCP oder UDP ausgelöst, kann mit Hilfe der ersten 8 Bytes des TCP- oder UDP-Protokollkopfes das Applikationsprogramm ermittelt und eine Fehlermeldung übergeben werden, z.B. daß die beim Verbindungsaufbauwunsch spezifizierte Server-Portnummer am Zielhost nicht aktiv ist.

Packet InterNet Groper - PING

Fehlermeldungen wie 'connection timed out' können eine ganze Reihe von Ursachen haben. Um erst einmal zu überprüfen, ob der ausgewählte Rechner überhaupt aktiv ist, ist das PING-Programm eine einfache aber effiziente Variante.
Ping arbeitet auf der Basis von ICMP, es verschickt ICMP-Echo-Request-Aufrufe. Sofern das IP-Protokollmodul im gestörten Rechner noch aktiv ist, wird es eine Antwort ('ICMP Echo Response') zurückschicken und damit beweisen, daß der Rechner erreichbar und aktiv ist.
Ping sendet standardmässig jede Sekunde ein 64 Byte langes Paket an den angegebenen Rechner und misst die Zeit bis zum Eintreffen der Antwort. Beendet wird Ping üblicherweise durch ein Abbruchsignal über die Tastatur (Ctrl C oder Del), danach wird eine Statistik über die empfangenen und verlorenen Pakete und die mittlere gemessene Antwortzeit ausgegeben.

Die Zeitmessung ist insofern interessant, als man daran erkennen kann, ob möglicherweise eine zu lange Laufzeit zwischen den Rechnern oder eine Überbelastung des Zielrechners den eigentlichen Grund des Problems darstellt. Normale Antwortzeiten liegen im Bereich <=10 bis 20ms. Verlorene Pakete deuten darauf hin, daß entweder der Zielrechner oder die dazwischen-geschalteten Knoten Pakete wegwerfen müssen - sei es aufgrund von unzureichendem Pufferplatz, internen Staus oder wegen Uebertragungsfehlern.
Als Optionen kann Ping eine andere Paketlänge und die Anzahl der Pakete mitgegeben werden (Beispiel unten: 512 Byte, 3 Pakete). Mit großen Paketen (bspw. über 1500 Byte in einem Ethernet) kann auch die Fragmentierung der Nachricht getestet werden.
Hinweis: Für ersten Test lokales Loopback-Ping mit ‘ping 127..0.0.1’ absetzen.

Beispiel eines PING-Vorgangs:

```
$ ping 111.2.33.4   512   3

PING 111.2.33.4:   512  data  bytes
512 bytes from 111.2.33.4:   icmp_seq=0.   time= 20.ms
512 bytes from 111.2.33.4:   icmp_seq=1.   time= 10.ms
512 bytes from 111.2.33.4:   icmp_seq=2.   time= 10.ms

--- 111.2.33.4   PING   Statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round trip (ms)   min / avg / max   =   10 / 13 / 20

$
```

Transmisson Control Protocol - TCP (RFC 793)

TCP bildet im DoD-Standard (Department of Defense) die ***Transportschicht***, d.h., es gewährleistet eine ***Kontrolle von Ende-zu-Ende-Verbindungen***, sorgt also für den sicheren Transport von Daten durch das Netzwerk. FTP, SMTP und TELNET benutzen TCP. TCP ist der ***Schicht 4 des ISO/OSI***-Referenzmodells zuzuordnen und ***setzt direkt auf dem internet protocol (IP) auf***. Die wesentlichen ***Dienstleistungen, die TCP für die Anwendungsprozesse bereitstellt, sind Paket-Reihenfolgegarantie, Verlustsicherung, Zeitüberwachung einer Verbindung, Multiplexing, Flußsteuerung, transparenter Datentransport, gesicherter Verbindungsauf- und -abbau sowie Verbindungsorientierung***.

Mit verbindungsorientiert bezeichnet man einen Datenaustausch zwischen Rechnern, wenn die Daten des Absenders durch den Empfänger bestätigt werden. Wesentlich für die verbindungsorientierte Kommunikation ist der Auf- und Abbau einer Verbindung. Die für die Nachrichtenverbindung geschaltete Leitung bleibt für die Dauer der Kommunikation zwischen den Teilnehmern erhalten (wie z.B. beim Telefon). Eine verbindungsorientierte Kommunikation besteht dann, wenn Subsysteme eine logische Verbindung (Sitzung) aufgebaut haben und über längere Zeit über die darunterliegenden Schichten Informationen austauschen. Die unbestätigte Verbindung nennt man im Gegensatz zur bestätigten 'verbindungslos' (connectionless, siehe auch unter 'IP').

Multiplexing bedeutet, daß über eine Verbindung mehrere logische Kanäle (Sessions) geöffnet werden können (siehe auch unter 'Portnummern').

TCP - Protokollkopf :

<table>
<tr><td colspan="16">1 4 10 16 24 32</td></tr>
<tr><td colspan="10">Sender-Port</td><td colspan="6">Empfänger-Port</td></tr>
<tr><td colspan="16">Sequenznummer</td></tr>
<tr><td colspan="16">Quittungsnummer</td></tr>
<tr><td colspan="3">Daten-abstand</td><td colspan="4">Reserviert</td><td>URG</td><td>PSH</td><td>SYN</td><td>ACK</td><td>RST</td><td>FIN</td><td colspan="6">Fenstergrösse</td></tr>
<tr><td colspan="10">Prüfsumme</td><td colspan="6">Urgent-Zeiger</td></tr>
<tr><td colspan="14">Optionen</td><td colspan="2">Füllzeichen</td></tr>
</table>

Sender- und Empfänger-Portnummer: Zwei 16 Bit breite Felder die die Endpunkte an beiden Enden der virtuellen Verbindung bezeichnen. Die Bedeutung der Portnummer werden wir im nächsten Abschnitt näher besprechen.

Sequenz- und Quittungsnummer: Jeweils ein 32-Bit-Wort, das die Stellung der Daten innerhalb des während dieser Verbindung ausgetauschten Datenstroms angibt. Die Sequenznummer gilt in Senderichtung, die Quittungsnummer für die Anzahl von Bytes, die von der anderen Seite empfangen wurden. Jeder der beiden Partner einer TCP-Verbindung generiert beim Verbindungsaufbau eine anfängliche Sequenznummer, deren wichtigste Eigenschaft sein muß, daß sie sich innerhalb des Zeitraums, in dem sich ein Paket noch im Netz befinden kann (IP-Lebenszeit), nicht wiederholt. Diese Nummern werden beim Verbindungsaufbau ausgetauscht und gegenseitig quittiert. Beim Datentransfer wird die Sequenznummer vom Absender jeweils um die Anzahl der bereits gesendeten Bytes erhöht. In der Quittungsnummer wird vom Empfänger angegeben, bis zu welchem Byte die Nachricht ordnungsgemäß empfangen wurde. Der große Zahlenraum von 32 bietet ausreichend Schutz vor Verdoppelungen und Überlappungen, da Sequenznummern bei Überlauf wieder mit 0 beginnen.

Datenabstand: Angabe der Länge des TCP-Protokollkopfes in 32-Bit-Worten zur Ermittlung des Datenbeginns.

Flags: Mit den Bits in diesem Feld werden Aktionen im TCP-Protokoll ausgelöst. Stehen sie auf logisch '1', bedeutet das:

Name: Erläuterung:
URG Zeiger im Urgent-Feld ist gültig (siehe 'Urgent-Zeiger' weiter unten)
ACK Quittungsnummer ist gültig.
PSH Daten in diesem Segment sollen sofort der Anwendung übergeben werden: eine Quittung für dieses Segment bedeutet, daß alle Daten bis zu dieser Quittungsnummer beim Kommunikationspartner angekommen sind.
RST Rücksetzen der Verbindung oder Antwort auf ein ungültiges Segment.
SYN Verbindungsaufbauwunsch, muß quittiert werden.
FIN Einseitiger Verbindungsabbau und Ende des Datenstroms aus dieser Richtung, muß quittiert werden.

Fenstergröße: Enthält die Anzahl von Bytes, die der Empfänger in seinen Datenpuffern augenblicklich für diese Verbindung aufnehmen kann ('Receive Window'). Mit dieser Angabe steuert das Empfänger-TCP den Datenfluss, eine Fenstersgröße von 0 würde z.B. das Sender-TCP effektiv stoppen! Durch allmähliches Anheben der Fenstergrösse wird der Datenfluss wieder in Gang gebracht.

Prüfsumme: Summiert Protokollkopf und Daten. Wie in allen anderen hier beschriebenen Protokollen wird zur Erstellung der TCP-Prüfsumme der gleiche Algorithmus wie für die IP-Prüfsumme verwendet.

Urgent-Zeiger: Ergibt zusammen mit der Sequenznummer einen Zeiger auf ein Datenbyte. Das damit definierte Datenbyte ist das Ende eines Nachrichtenabschnitts, die danach folgenden Daten werden auf diese Weise als wichtig gekennzeichnet. Diese Funktion wird als 'Urgent Data' bezeichnet.

Optionen: In TCP gibt es nur drei Optionen: 'End of Option', 'List No Operation' und 'Maximum Segment Size'. Letztere wird beim Verbindungsaufbau gesendet, um die Bereitschaft zum Empfang von größeren Segmenten als 536 Bytes anzuzeigen.

Um die oben aufgeführten Dienste zu gewähren stellt TCP ganz bestimmte Anforderungen an das Protokoll der darunterliegenden Vermittlungsschicht (OSI-Layer 3). Diese muß in der Lage sein, Datagramme über einen Verbund von Netzen hinweg zu senden, die Partner eindeutig zu adressieren, Datenpakete nach den jeweiligen Netzkonventionen zu zerlegen und zu reassemblieren sowie Informationen über die Paketreihenfolge und Sicherheitsmerkmale zu übermitteln. Diese Leistungen werden alle vom Internet Protocol (IP) erbracht.
Die TCP/IP-Protokollfamilie wurde schon vor ca. 30 Jahren vom DoD entwickelt. Ziel war die Schaffung möglichst Code-kompakter Protokolle für die Router. Danach hat sich lange Zeit niemand mehr für TCP/IP interessiert, da man hoffte, auf Basis von OSI-Protokollen einen Verbund heterogener Systeme realisieren zu können. Diese Hoffnung trog, und die mittlerweile auf dem Unix-Sektor weiterentwickelten Protokolle wurden schnell für die Vernetzung unterschiedlichster Systeme populär. Heute findet man kein professionelles System mehr, das nicht mit TCP/IP-Protokollen ausgestattet werden kann. Somit ist die DoD-Protokollfamilie in vielen gemischten Umgebungen heute oft die einzige gemeinsame Möglichkeit der Kommunikation.

Portnummern

TCP verwendet sogenannte Portnummern (Kanalnummern). Die Portnummern sind 16 Bit breite Felder, die die ***Endpunkte an beiden Enden einer virtuellen Verbindung bezeichnen*** (adressieren). Da ***Portnummern 16 Bit*** gross sind, könnte ein Hostrechner theoretisch gleichzeitig ***bis zu 65535 verschiedene TCP-Verbindungen*** aufbauen.
Auch UDP verwendet zur Adressierung Portnummern, zu beachten ist aber, daß TCP und UDP jeweils eigene Adreßräume haben, d.h. Portnummer 501 in TCP ist nicht identisch mit Portnummer 501 in UDP. Der Gültigkeitsbereich einer Portnummer ist auf einen Host beschränkt. ***Netzwerknummer, Host-ID und Portnummer*** spezifizieren zusammen einen ***Kommunikationsendpunkt, auch Socket genannt***.
Ähnlich wie beim Telefon geht auch der Aufbau einer Verbindung über TCP vonstatten: auch hier gibt es einen passiven Partner, den Angerufenen, und einen aktiven Partner, den Anrufer. Bevor zwei Programme miteinander kommunizieren können, müssen beide Kommunikationsendpunkte eröffnen, deren Adressen in den Protokollköpfen der einzelnen Schichten verwendet werden. Damit sie allerdings zur Verbindungsaufnahme zusammenfinden, muß die Adresse des passiven Partners dem aktiven Partner bekannt sein, d.h. beide Seiten müssen vorher eine Portnummer vereinbart haben, unter der der passive Partner, auch Server genannt, auf den Verbindungsaufbau wartet. Die Portnummer des aktiven Partners, auch Client genannt, ist irrelevant, solange der Server keine spezielle Portnummer für den Client vorschreibt.
Wie im Telefonsystem die Störungsstelle oder Zeitansage, könnte man TCP/IP-Applikationen, wie z.B. Telnet oder FTP, als feste Dienstleistungen betrachten, die unter einer definierten, allgemein bekannten Nummer erreichbar sind. Eine derartige definierte Nummer wird in der TCP/IP-Welt als ***well-known port number*** bezeichnet. Man kann daher sagen:

> Jeder Dienst (Service) muß eine eigene Portnummer festlegen. Über diese Portnummer adressiert der Kunde (Client) den Server, der den Dienst bereitstellt.

In untenstehender Tabelle finden Sie als Beispiele einige der auf UNIX-Rechnern verfügbaren Dienste und deren Portnummern sowie das Protokoll, unter dem diese Dienste arbeiten.

Dienst	Portnummer	Protokoll
TCP	6	
TELNET	23	TCP
FTP	21	TCP
SMTP	25	TCP
HTTP	80	TCP
rlogin	513	TCP
UDP	17	
TFTP	69	UDP
rwhod	513	UDP

Sockets

Sockets sind die Kombination aus einer IP-Adresse und der dazugehörigen Portnummer. Der Socket ist der abstrakte Endpunkt für die Kommunikation. Die Konzeption der Sockets stammt von auf Berkeley basierenden UNIX-Systemen. In Berkeley-UNIX ist der Socket eine auf I/O basierende Konzeption und ein Endpunkt im Kommunikationsprozess. (Bei Microsoft spricht man in diesem Zusammenhang von Win-Sockets.)

TCP/IP - Protokollszenarien

In diesem Abschnitt zeige ich in drei Abläufen, wie TCP die Felder im TCP-Protokollkopf in den wichtigsten Phasen einer Verbindung verwendet - dem ***Verbindungsaufbau***, dem ***Datenaustausch*** und dem ***Verbindungsabbau***. In den Abbildungen zeigen die Pfeile jeweils die Senderichtung des Segments. Die Klammern umgeben die Felder des TCP-Protokollkopfes.

Bild 1 - **Verbindungsaufbau in TCP**

TCP A		TCP B
→	(SEQ=100), (FLAGS=SYN)	
	(SEQ=300) ,(ACK= 101) ,(FLAGS=SYN, ACK)	←
→	(SEQ=101), (ACK=301), (FLAGS=ACK)	

Abbildung 1 zeigt einen Verbindungsaufbau. Dabei findet ein sog. ***three-way handshake*** statt, in dem jede Seite, ausgelöst durch das SYN-Flag, die Sequenznummer der anderen Seite durch Erhöhen um 1 quittieren muß. Die Sequenznummer (SEQ=100) von TCP A findet sich in der Antwort von TCP B im Quittungsfeld (ACK=101) wieder und umgekehrt. Das ACK-Flag zeigt jeweils an, daß der Wert im Quittungsfeld gültig ist.

Bild 2 - **Datenaustausch in TCP**

TCP A		TCP B
→	(SEQ=l01), (ACK=301), (FLAGS=ACK), (DATA=5)	
	(SEQ=301) ,(ACK= 106) ,(FLAGS=ACK) ,(DATA=10)	←
→	(SEQ=106), (ACK=311), (FLAGS=ACK)	

In Abbildung 2 sehen wir einen Datenaustausch, der gleichzeitig in beiden Richtungen stattfindet. Zu erkennen ist hier, wie die Empfängerseite jeweils die im SEQ-Feld eingetragene Sequenznummer (SEQ=101) um die Anzahl der empfangenen Datenbytes (DATA=5) im ACK-Feld (ACK=106) quittiert. Zusammen mit der Quittung werden in der anderen Richtung Daten gesendet, die ihrerseits auf dieselbe Art und Weise quittiert werden.

Bild 3 - **Verbindungsabbau in TCP**

TCP A		TCP B
→	(SEQ=l06), (FLAGS=FIN, ACK)	
	(SEQ=31l), (ACK=107), (FLAGS=ACK)	←
	(SEQ=311) ,(ACK=107) ,(FLAGS=ACK) ,(DATA=5)	←
→	(SEQ=107), (ACK=316), (FLAGS=ACK)	
	(SEQ=316) ,(ACK=107) ,(FLAGS=FIN, ACK)	←
→	(SEQ=107), (ACK=317), (FLAGS=ACK)	

In Abbildung 3 wird durch Senden eines FIN-Flags von TCP A die Verbindung einseitig abgebaut. TCP B quittiert das FIN-Flag durch Erhöhung der Sequenznummer (SEQ=106) im Antwortsegment um 1 (ACK=107). Damit ist der Erhalt aller bis zu diesem Zeitpunkt von TCP A gesendeten Daten quittiert. Danach dürfen keine Daten mehr von TCP A abgesendet werden. TCP B kann jedoch weiterhin senden, bis es seinerseits ein FIN abschickt.

Während im Verbindungsaufbau eine Seite aktiv und die andere passiv ist, herrscht während des Datenaustauschs und in der Initiative zum Verbindungsabbau vollkommene Gleichberechtigung zwischen beiden Partnern.

User Datagram Protocol - UDP (RFC 768)

Das user datagram protocol ist ein ***Transportprotokoll (Schicht 4) des ISO/OSI-Modells*** und unterstützt den ***verbindungslosen Datenaustausch*** zwischen Rechnern. Das UDP wurde definiert, um auch Anwendungsprozessen die direkte Möglichkeit zu geben, Datagramme zu versenden. UDP baut direkt auf dem darunterliegenden Internet-Protokoll auf. Im Gegensatz zu TCP ist UDP ein verbindungsloses Transportprotokoll. UDP ist, verglichen mit TCP, ein regelrechtes Leichtgewicht:
UDP garantiert weder die Ablieferung eines Datagrammes beim Zielpartner, noch sind Vorkehrungen gegen eine Duplizierung oder eine Reihenfolgevertauschung getroffen. Allerdings macht das Fehlen dieser Zusätze UDP besonders effizient und daher geeignet für Hochgeschwindigkeitsanwendungen wie z.B. verteilte Dateisysteme (NFS) oder dergleichen, die ohnehin nur auf schnellen und sicheren Übertragungswegen wie z.B. in LANs sinnvoll verwendet werden.
Durch Verwendung von sogenannten 'Ports' können in einem Rechner mehrere Kommunikationsziele ausgewählt werden. Diese Funktion wird 'Multiplexing' genannt. Die Ports werden durch Nummern identifiziert, wobei diese Nummern teilweise für bestimmte Prozesse reserviert sind (siehe oben 'Portnummern').

UDP-Protokollkopf:

1 16	17 32
Sender-Portnummer	Empfänger-Portnummer
Länge	Prüfsumme

Sender- und Empfänger-Portnummer: Wie im TCP-Protokoll sind auch hier die Portnummern die Referenz zu den Transportprotokollbenutzern.

Länge: Enthält die Länge des gesamten Datagramms, inklusive des Protokollkopfs.

Prüfsumme: Enthält die Internet-Prüfsumme der Daten und des Protokollkopfs. Wenn dieses Feld den Wert 0 enthält, hat der Absender keine Prüfsumme eingetragen und es findet im Empfänger-UDP keine Prüfung statt.

TCP/IP in Weitverkehrsnetzen

TCP/IP Routing

Im Zusammenhang mit ***TCP/IP-Routing*** werden eine Reihe von Protokollen eingesetzt, die der Endsystem-Router- und Router-Router-Kommunikation dienen: ***IP (internet protocol), RIP (routing information protocol), ICMP (internet control message protocol), ARP (address resolution protocol) und EGP (exterior gateway protocol***). Es soll hier als Abrundung nur ein Überblick über das RIP gegeben werden.

Router Information Protocol - RIP (RFC 1058)

Beim Routing arbeitet man mit Tabellen, die bei statischem Routing manuell angelegt werden, bei dynamischem Routing von den Routern erlernt und danach angelegt werden. RIP wurde auf der Basis des XNS RIP entwickelt und hat sich als Standardmodul des BSD Unix 4.x sehr stark etabliert. Viele Internet-Netze haben es übernommen. ***Bei RIP schicken alle Router in Intervallen ihre eigenen Routing-Tabellen als Broadcast an die anderen Router***. Die Entfernung zu anderen Netzwerken wird dabei in Relation, d.h. aus der Sichtweise der eigenen Routing-Tabelle angegeben. Auf der Basis der empfangenen Tabellen berechnen die Router die kürzesten übermittelten Entfernungen zu jedem Zielnetz und nehmen den Nachbar-Router, der diese Entfernung bekanntgegeben hat, als Ziel-Router zur Weiterleitung. Die maximale Entfernung darf 14 Hops betragen, der Wert 15 steht für 'nicht erreichbar'.

TCP/IP über serielle Leitungen

Um unabhängig vom Ort jederzeit unbeschränkten Zugang zum Internet erlangen zu können, ist es notwendig, auch über Telefonleitungen oder andere serielle Datenwählleitungen das Internet Protocol (IP) benutzen zu können. Dies kann mit Hilfe des Serial Line Internet Protocol (SLIP) oder mit dem Point-to-Point (PPP) Protocol geschehen.

Serial Line Internet Protocol - SLIP

Das Serial Line Internet Protocol (SLIP) wurde 1984 von Rick Adams für die Berkeley Unix Version 4.2 entwickelt. Es wird gewöhnlich dazu benutzt, das Internet-Protokoll über Wählleitungen mit Geschwindigkeiten zwischen 1200 bit/s und 19,2 kbit/s zu betreiben. Der dazu benutzte Algorithmus ist denkbar einfach. SLIP benötigt lediglich zwei Steuerzeichen, nämlich das END-Zeichen (Oktal 300 bzw. Dezimal 192) sowie das ESC-Zeichen (Oktal 333 bzw. Dezimal 219) (nicht zu verwechseln mit dem ASCII-Escape-Zeichen!).
Um nun ein IP-Paket zu versenden, wird begonnen, die Daten (ohne Header) zu übertragen. Am Ende des Paketes wird ein SLIP-END-Zeichen hinzugefügt. Tritt innerhalb der zu übertragenden Daten zufällig die Bitfolge des END-Zeichens selbst auf, so werden vor dieses zwei Bytes mit ESC und ein Byte mit der Bitfolge für Dezimal 220 eingefügt. Enthalten die Daten die Bitfolge des ESC-Zeichens, so werden zwei Bytes mit ESC und ein Byte mit der Bitfolge für Dezimal 221 eingefügt.

Einige SLIP-lmplementationen fügen vor die Übertragung des Datenpaketes ein END-Zeichen ein, um etwaige Fehlerbytes, verursacht durch Leitungsstörungen, zu beseitigen. Da SLIP kein offizieller Standard ist, gibt es auch keine eindeutig definierte maximale Paketlänge. Als Richtlinie wird im allgemeinen die SLlP-lmplementation für das Berkeley-UNlX betrachtet, die eine maximale Paketlänge von 1006 Bytes festlegt (ohne END und ESC Zeichen).

Aufgrund seiner Einfachheit ist SLIP sehr leicht zu implementieren und auch sehr weit verbreitet. Es gibt jedoch einige wesentliche funktionelle Einschränkungen. So können keine Adreßinformationen über SLIP ausgetauscht werden. Beiden Computersystemen muß daher von vornherein die jeweilige andere IP-Adresse bekannt sein, um kommunizieren zu können.
SLIP besitzt desweiteren kein Protokoll-Typenfeld. Es kann daher nur ein Protokoll zur Zeit übertragen werden. Die Verbindung zweier Multiprotokoll-Systeme wird damit unmöglich.
Und schließlich ist in SLIP keinerlei Fehlerkorrekturmechanismus integriert. Dies wird dann zwar vom IP- bzw. TCP-Protokoll übernommen, es wäre jedoch effizienter, wenn bereits auf unterster Ebene ein einfacher Korrekturmechanismus existieren würde. So müssen im Fall einer Übertragungsstörung meist mehrere Pakete übertragen werden, bis der IP- bzw. TCP-Fehlerkorrekturmechanismus eingreifen kann.

Point-to-Point Protocol - PPP (RFC 1331)

Auch das Point-to-Point-Protokoll ermöglicht die Übertragung über eine serielle Datenleitung. Es ist wesentlich komplexer als SLIP, und hat im Gegensatz zu diesem eine Vielzahl von Funktionen wie Fehlerkorrektur oder die Möglichkeit der Zuweisung von Netzwerkadressen.

PPP besteht aus drei Komponenten: Der Methode zur ***Enkapsulierung*** der Datenpakete, dem ***Link-Control-Protokoll*** zum Aufbau der Datenverbindung, sowie verschiedenen ***Netzwerk-Kontrollprotokollen*** zu Konfiguration der Parameter für die unterschiedlichen Netzwerkprotokolle (IP, IPX, DECnet etc.).

- ***PPP-Enkapsulierung***

PPP enkapsuliert die zu übertragenden Datenpakete in Standard HDLC (High-level Data Link Control ISO 3309-1979)-Format. (Bei asynchroner Übertragung werden von HDLC zusätzlich vor und nach jedem Byte ein Start- bzw. Stop-Bit eingefügt.)

- ***PPP Link Control Protocol (LCP)***

Das LCP-Protokoll hat die Aufgabe, die serielle Punkt-zu-Punkt-Verbindung aufzubauen, zu überwachen, Netzwerk-Kontrollprotokolle aufzurufen, sowie nach dem Ende der Datenübertragung die Verbindung wieder abzubauen. Werden in einem PPP-Paket LCP-Daten übertragen, so ist dieses am Protokoll-Typenfeld ‘hex c021 Link Control Protocol’ erkennbar. Nachdem ein Austausch von LCP-Konfigurationspaketen erfolgt, und ein ‘Configure-Ack’-Paket von der Gegenstelle erhalten wurde, wird die Datenleitung geöffnet.
Nun kann (optional) die Leitungsqualität durch das Versenden von LCP Echo-Request bzw. LCP Echo-Reply-Paketen getestet werden. Nach dieser Testphase ruft das Link Control Protocol schließlich das gewünschte Netzwerk-Kontrollprotokoll-Programm (NCP Network Control Protocol) auf.

- ***PPP-Netzwerk-Kontrollprotokoll für das IP (IPCP)***

Für jedes Protokoll, dessen Übertragung von PPP unterstützt wird, gibt es ein Netzwerk-Kontrollprotokoll. Die entsprechende Implementation für das Internet-Protokoll nennt sich lPCP (IP Control Protocol). Die Aufgabe des IPCP ist es, die IP-Module auf den beiden kommunizierenden Computersystemen zu aktivieren bzw. zu deaktivieren. IPCP wird in PPP-Paketen mit dem Protokoll-Typenfeld ‘hex 8021 Internet Protocol Control Protocol’ übertragen. Nachdem auch das IPCP die Datenleitung auf Netzwerk-Ebene geöffnet hat, können die eigentlichen IP-Pakete übertragen werden.

Anwendungsprogramme auf TCP/IP-Basis

Bisher haben wir uns mit den Protokollen der unteren Schichten beschäftigt. Zur TCP/IP -Architektur gehört aber auch ein Satz standardisierter Applikationsprotokolle. Auch wenn diese in der neusten Zeit teilweise durch modernere und den gewachsenen Ansprüchen mehr entsprechende Dienste und Applikationen ersetzt worden sind, lohnt es sich, mehr über sie zu erfahren.

Telnet (RFC 854)

TELNET soll dazu dienen, Zugriff auf einen am Netz angeschlossenen Rechner in Form einer ***Terminalsitzung, auch remote login oder virtual terminal*** genannt, zu liefern. Telnet ermöglicht eine bidirektionale Kommunikation, um Datenendgeräte mit entsprechenden Prozessen zu verbinden. Das Zielsystem wird dabei allgemein als Server oder Host bezeichnet, das eigene lokale System als Client. Auf UNIX-Systemen läuft auf der Clientseite das Kommando telnet, auf der Serverseite, d.h. auf dem über telnet angewählten Rechner, ein sogenannter Daemon oder Server namens telnetd. Der ***TELNET-Dienst*** wurde auf ***TCP-Port 23*** gelegt.

Auf UNIX-Rechnern wird heutzutage das Kommando rlogin mitgeliefert, das in etwa die gleiche Funktionalität wie telnet bietet, jedoch die UNIX-Umgebung besser unterstützt. Das TELNET-Protokoll enthält allerdings Elemente, die insbesondere die Zusammenarbeit mit Großrechnersystemen unterstützen und ist deshalb für den heterogenen Betrieb vorzuziehen.

- TELNET-Sitzung

Der Aufruf von telnet erfolgt in der Regel mit der Angabe eines Rechnernamens. Zusätzlich kann eine Portnummer angegeben werden: auf diese Weise lässt sich ein Dialog mit anderen Diensten wie z.B. SMTP manuell abwickeln. Wird kein Rechnername angegeben, geht telnet in einen Kommandomodus und liefert eine Eingabeaufforderung. In diesem Kommandomodus lassen sich Verbindungen zu Rechnern aufbauen und abbrechen, die Betriebsparameter beeinflussen und vieles mehr. Durch ein Fluchtzeichen- standardmässig ^] (d.h. gleichzeitiges Drücken der Tasten Ctrl und]) - kann man jederzeit in diesen Modus überwechseln. Allerdings wechselt telnet bei aktiver Verbindung nach jedem Kommando wieder auf die Netzverbindung zurück.
Durch den Start von telnet wird eine TCP-Verbindung zum Zielrechner und zum telnetd (also dem Server) aufgebaut. Telnetd arrangiert dann den weiteren Ablauf derart, daß das Login, so wie man es normalerweise an einem lokalen Terminal gewohnt ist, nun über die gerade entstandene Verbindung vonstatten geht - sozusagen über ein ‘künstliches’ Terminal. Das Ergebnis ist ein Fenster zum Server-Rechner, in dem eine Shell läuft, über die man wie gewohnt Programme starten kann.

- Arbeitsweise des TELNET-Protokolls

Wie Sie sicher wissen, ist die Bedienung von Computersystemen sowie die Fähigkeiten der daran angeschlossenen Terminals von System zu System sehr unterschiedlich. Erst mit Standard Betriebssystemen wie UNIX, DOS oder Windows ist dieser Bereich für den Benutzer einigermaßen vereinheitlicht worden. Ein Protokoll, das in der sehr heterogenen DV-Welt zwischen beliebigen Systemen funktionieren soll, muß daher entweder sehr restriktiv hinsichtlich der zugelassenen Funktionen oder sehr flexibel in der Anpassung an bestehende Verhältnisse sein.

Im TELNET-Protokoll hat man sich für einen Mittelweg entschieden. Es besteht aus einigen wenigen Festlegungen hinsichtlich des Formats von Nachrichten und darüber hinaus der Möglichkeit, alles weitere zwischen den an der Sitzung beteiligten Partnern zu verhandeln.

- Network Virtual Terminal
Zur Beschreibung der Fähigkeiten der Datenquellen und -empfänger definiert TELNET ein sogenanntes ***Network Virtual Terminal (NVT).*** Das NVT besteht aus einer virtuellen Tastatur, die nur bestimmte Zeichen erzeugen, und einem virtuellen Drucker, der nur bestimmte Zeichen anzeigen kann. Damit ist der Funktionsumfang der Ein- und der Ausgabeseite festgelegt. Jeder Endpunkt einer TELNET-Verbindung besteht logisch aus einer NVT-Tastatur und einem NVT-Drucker. Genaue Angaben finden Sie in der TELNET-Spezifikation. Das NVT ist damit gewissermaßen eine Spezifikation der Darstellungsschicht und deshalb von weitreichender Bedeutung. Neben TELNET wird das NVT-Format auch im FTP- und im SMTP-Protokoll verwendet.

Beispiel einer Telnet-Sitzung:

```
$ telnet
telnet> toggle options
Will show option processing
telnet> open BKWHOST
Trying ...
Connected to BKWHOST.
Escape character is‘^]’.
SENT do SUPRESS GO AHEAD
SENT will TERMINAL TYPE (reply)
RCVD do TERMINAL TYPE (don’t reply)
RCVD will SUPRESS GO AHEAD (don’t reply)
RCVD will ECHO (reply)
SENT will ECHO (reply)
RCVD do ECHO (reply)
SENT wont ECHO (reply)

System V UNIX (BKWHOST)

RCVD dont ECHO (don’t reply)
login: testaccount
Password:
UNIX System V release 3.5
BKWHOST
Copyright © 1990 AT&T
$ date
Sun Apr 04 11:22:33 GMT 1996
$ ^]
telnet> quit
Connection closed by foreign host.
$
```

File Transfer Protocol - FTP (RFC 959)

Das File Transfer Protokoll (FTP) legt ***TCP-Port 21 als Kommandokanal und TCP-Port 20 als Datenkanal*** fest. Auf UNIX-Systemen existiert zum Protokoll das gleichnamige Kommando ftp sowie der Server ftpd.
FTP unterscheidet sich in mancher Hinsicht von anderen Dateitransfer-Programmen. Zu den herausragendsten Unterschieden zählen die Verwendung von ***getrennten Kanälen für Kontrollinformationen und Daten*** sowie der Umstand, daß FTP-Datentransfers nicht im Hintergrund ablaufen, d.h. ohne einen Spooler arbeiten.

FTP verwendet als Protokoll-Elemente durch Newline-Zeichen terminierten ASCII-Text im NVT-Format, der aus einem vier Zeichen langen Kommandowort mit optionalen Parametern besteht. ***Rückmeldungen*** beinhalten einen ***dreistelligen Zifferncode*** und einen erklärenden Text, der eine Erläuterung zum Erfolg oder Mißerfolg der Aktion gibt. Kodes beginnend mit den Ziffern 1, 2 oder 3 sind positive Rückmeldungen, die Kodes 4 oder 5 stehen für Fehler. Die darauf folgenden Ziffern definieren die Rückmeldung bzw. den Fehler näher. Weil die Protokoll-Elemente in ASCII kodiert sind, wird die Fehlersuche vereinfacht, und man kann auch interaktiv mit dem FTP-Server kommunizieren. Die FTP-Protokollkommandos erlauben das Senden, Empfangen, Löschen und Umbenennen von Dateien, das Einrichten, Löschen und Wechseln von Dateiverzeichnissen, das Anfügen an Dateien sowie das Versenden von elektronischer Post und vieles mehr.

Für jeden Datentransfer wird eine TCP-Verbindung zwischen Client und Server eröffnet und nach der Übertragung wieder geschlossen. Auf diese Art und Weise verwendet FTP die Sicherungsfunktionen von TCP- das ja bereits alle nötigen Vorkehrungen zur fehlerlosen Übertragung trifft - und muß darüber hinaus keine eigenen Sicherungsfunktionen anwenden.
Obwohl mehrere Übertragungsmodi wie z.B. die Komprimierung von Daten spezifiziert sind, werden auf UNIX-Systemen nur zwei Modi implementiert: Text- und Binär-Modus. Im Textmodus werden Textdateien als durch Carriage-Return und Newline getrennte ASCII-Zeilen versendet und lassen sich somit zwischen zwei unterschiedlichen Systemen transferieren. Im Binärmodus wird eine Datei ohne jegliche Umwandlung als eine Folge von Bytes übertragen, was natürlich wesentlich schneller geht. Nach dem Start ist zunächst der Textmodus eingestellt - solange ein Übertragung zwischen gleichartigen Systemen stattfindet, sollte man aus Gründen der Geschwindigkeit aber auch für Textdateien den Binärmodus verwenden.

- FTP-Kommando
Zu Beginn einer FTP-Sitzung muß eine Verbindung zum Server-FTP aufgebaut werden. Das geschieht durch die Angabe des Rechnernamens im ftp-Kommando oder durch das open-Kommando im FTP-Kommandomodus, ähnlich wie bei telnet. Um einen Dateitransfer auszuführen, muß zudem die Berechtigung zum Zugriff auf die Datei oder die Erlaubnis zum Abspeichern geholt werden. Dies geschieht durch Anmelden auf dem Serverrechner unter der Angabe einer Benutzerkennung und eines Paßworts. Nach dem Login befinden man sich jedoch nach wie vor logisch auf dem Client und nicht- wie bei TELNET - auf dem Serverrechner. Dies ist zu beachten, wenn Kommandos ausgeführt werden.
Jetzt kann eines der vielen FTP-Kommandos angestoßen werden. In meinem Beispiel unten ist die Debug (-d) Option eingeschaltet, um zu zeigen, was auf dem Kommandokanal ausgetauscht wird. Wie man feststellen kann, ist der Ablauf einer Sitzung sehr einfach mitzuverfolgen. Pfeile zeigen eine Nachricht des Clients an, Zeilen mit dreistelligen Zahlen am Beginn sind Antworten des Servers. Der Transfer der Daten kann leider nicht auf diese Weise mitverfolgt werden. Die PORT-Aufrufe signalisieren dem Server die Adresse eines Datenkanals, an dem der Client wartet. Fehlt der PORT-Aufruf, wird standardmäßig TCP-Port 20 verwendet.

Ein Nachteil von FTP ist, daß dieses Programm die Zugriffsberechtigungen der Quelldatei nicht auf die Zieldatei übernehmen kann - dazu fehlen leider die protokolltechnischen Voraussetzungen. Außerdem können keine Dateibäume übertragen werden.

Beispiel einer FTP-Sitzung:

```
$ ftp -d
ftp> open BKW2
Connected to BKW2.
220 BKW2 FTP server (Version 4.4 Tue Dec 20 1988) ready.
Name (BKW2:BORl):BORl
331 Password required for BORl.
Password: ....
230 User BORl logged in.
ftp> dir
-> PORT 192,9,150,244,4,57
200 PORT command successful.
-> LIST 150 Opening data connection for /bin/ls -l (.....
total 1
-rw------   1 BORl    240 May 18 10:53 test
226 Transfer complete.
100 bytes received in 1 seconds (0.66 Kbytes/s)
ftp> get test
-> PORT 192,9,150,244,4,59
200 PORT command successful .
-> RETR test
150 Opening data connection for test (192.9.150.244,1083)
226 Transfer complete.
local: test remote: test
250 bytes received in 0 seconds (0.24 Kbytes/s)
ftp> quit
-> QUIT
221 Goodbye.
```

Die folgenden ftp-Kommandos werden häufig verwendet:

ftp-Kommando	Beschreibung
open	Verbindung aufbauen
get	Holen einer Datei vom Server
put	Senden einer Datei an den Server
del	Löschen einer Datei im Server
binary	Umschalten in den binären Übertragungsmodus
cd	Wechsel des Dateiverzeichnisses auf dem Server
lcd	Wechsel des Dateiverzeichnisses auf dem Client
pwd	Ausgeben des Dateiverzeichnisses auf dem Server
quit	Beenden des ftp-Kommandos

Trivial File Transfer Protocol - TFTP (RFC 783)

Das Trivial File Transfer Protocol ist ein ***Dateitransfer-Protokoll für Minimalanforderungen***. Es verwendet den UDP-Port 69.
Wie FTP unterstützt auch TFTP einen Text- und einen Binär-Übertragungsmodus. Hauptmerkmal gegenüber FTP ist aber die Verwendung eines verbindungslosen Transport-Protokolls, in unserem Fall also UDP. Für die Gestaltung des Protokolls hat das mehrere Konsequenzen: Zunächst muß TFTP selbst die Sicherung der Übertragung durch Algorithmen wie Zeitüberwachung und Paket-Wiederholung vornehmen. Außerdem wird kein 'Einloggen' auf dem Serverrechner durchgeführt, der TFTP-Server ersetzt die fehlende Autorisierung durch restriktive Zugriffsbeschränkungen: so dürfen z.B. auf UNIX-Rechnern nur solche Dateien gelesen und geschrieben werden, deren Zugriffsrechte sie von allen Benutzern lesbar und beschreibbar machen. In welcher Weise ein System sich gegen unberechtigte Zugriffe über TFTP schützt, ist allerdings nicht festgelegt und deshalb implementierungsspezifisch.
Die Vorteile von TFTP liegen sicher nicht im Bereich des regelmässigen Dateitransfers zwischen Systemen. Obwohl TFTP dazu in Ausnahmefällen verwendet wird, liegt das Hauptanwendungsgebiet dieses Protokolls heutzutage vor allem im Laden von Serverprogrammen und Fonts in X Window-Terminals sowie im Bootstrapping von plattenlosen Workstations. Im letzteren Fall wird TFTP für den Transfer des Systemprogramms in den Hauptspeicher verwendet. Grund dafür sind die geringen Voraussetzungen zum Betrieb von TFTP: außer dem Protokoll selbst benötigt man lediglich die Basisfunktionen von IP, das sehr einfache UDP-Protokoll sowie einen Treiber für den Zugang zum Netzwerk. Eine solche Protokollsäule ist mit geringstem Aufwand zu implementieren: das Resultat hat in wenigen Kilobyte Speicher Platz - wie z.B. einem EPROM.
Zum TFTP-Protokoll gibt es das gleichnamige tftp-Kommando. Die verfügbaren Optionen und die Bedienung sind dem ftp-Kommand sehr ähnlich, allerdings weniger umfangreich, so entfällt z.B. der Authorisierungsvorgang. Beim Beispiel unten stellt die Anweisung 'connect' keine Verbindung her, sondern legt die Partneradresse für die folgenden Aktionen fest (tftp ist verbindungslos). 'Verbose' und 'trace' machen den Austausch der Protokollelemente sichtbar.

Beispiel einer TFTP-Sitzung:

```
$ tftp
tftp> connect bkw
tftp> binary
tftp> trace
Packet tracing on.
tftp> verbose
Verbose mode on.
tftp> get /user1/datei /user2/datei
getting from bkw:/user1/datei to /user2/datei [octet]
sent RRQ <file=/user1/datei, mode = octet>
received DATA <block = 1, 512 bytes>
sent ACK <block = 1>
received DATA block = 4, 325 bytes>
Received 837 bytes in 0.1 seconds [ 66960 bits/sec]
tftp> quit
$
```

Simply Mail Transfer Protocol - SMTP (RFC 821)

SMTP ist der Internet-Standard zur Verteilung von elektronischer Post. Er ist ***textorientiert*** und setzt auf TCP auf (***TCP-Port 25***). Eine Nachricht besteht aus Kopf und Rumpf. Der Kopf enthält u.a. Datum, Bezug, Empfänger, Absender, Kopienempfänger; der Benutzer wird für jeden dieser Einträge durch einen 'prompt' angesprochen. Der Rumpf besteht typischerweise aus freiem ASCII-Text. Nachrichten mit mehreren Empfängern auf einem Ziel-Host werden nur einmal zum Ziel übertragen und dort verteilt. Im Vergleich zu X.400 handelt es sich bei SMTP um ein funktional ärmeres System.

Beispiel eines Mail-Dialoges:

```
220 AI.BKW.CH Sendmail 5.1 ready at Sat, 10 Jun ..
HELO FTS.PTT.CH
250 AI.BKW.CH Hello FTS.PTT.CH. pleased ..
MAIL From: <BOR@AI>
250 <BOR@AI>... Sender ok
RCPT To: HEGI
250 HEGI... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
Date: 10 Jun 89 11:22:33
From: <BOR@ AI.BKW.CH>
To: <HEGI@FTS.PTT.CH> Subject: Alles paletti
SMTP läuft wunderbar!
.
250 Ok
QUIT
221 AI.BKW.CH closing connection
```

Erläuterung der SMTP-Kommandos:

SMTP-Befehl	Beschreibung
HELO	Vorstellung des Clients beim Server.
DATA	Text der Nachricht im ASCII-Format (7Bits). Das Ende der Nachricht wird durch einen einzelnen Punkt am Anfang einer Zeile angezeigt.
QUIT	Beenden des SMTP-Dialogs.
MAIL	Angabe des Absenders der Nachricht. An ihn werden Antworten und Fehlermeldungen geschickt.
RCPT	Angabe des Empfängers der Nachricht. Es können mehrere RCPT-Kommandos hintereinander folgen: damit werden die Daten an mehrere Empfänger verschickt, aber nur einmal übertragen.
SEND	Senden der Nachricht an das Terminal eines Benutzers.
HELP	Anforderung von Hilfe-Informationen.
RSET	Abbruch der augenblicklichen Mail-Transaktion.

Als Antworten und Quittungen werden die dreistelligen Codes vom FTP-Protokoll verwendet.

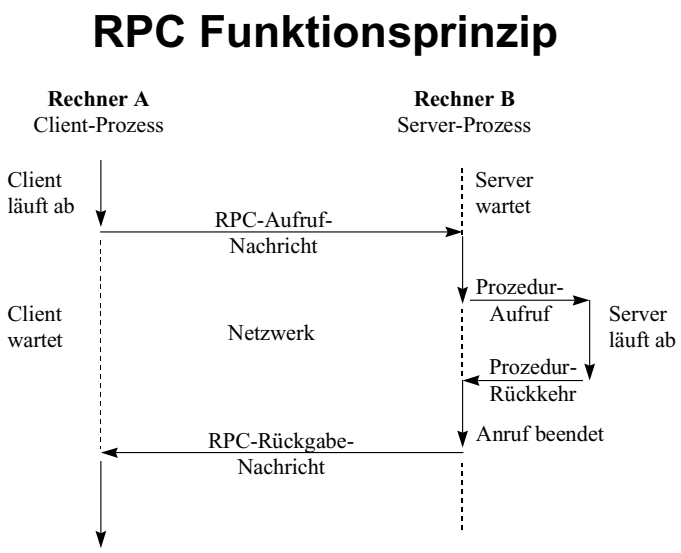
Remote Procedure Call - RPC (RFC 1050)

RPC ist auf der Kommunikations-Steuerungsschicht (Session Layer) angesiedelt und gewährleistet einen ***entfernten Funktionsaufruf***. Jeder Server im Netz stellt im Rahmen dieses Konzeptes eine Anzahl von Services zur Verfügung, die mit RPC angefordert werden können. Diese Funktionen sind als Prozeduren eines Programmes realisiert und können unter Angabe von Serveradresse, Programmnummer und Prozedurnummer angesprochen werden. RPC wurde ursprünglich für NFS entwickelt, wird aber auch in vielen vernetzten Applikationen verwendet. Der Quellcode von RPC ist über Sun Microsystems unentgeltlich verfügbar.

Das Funktionsprinzip ist dem lokaler Prozeduraufrufe ähnlich. Im Falle eines lokalen Prozeduraufrufs werden die Parameter an die aufzurufende Prozedur an einer bekannten Stelle wie z.B. dem Stack hinterlegt. Danach verzweigt das Programm in die Unterprozedur, die ihre Aufgaben abwickelt und wieder zurückkehrt. Die von der Unterprozedur erzeugten Resultate können nun von der aufrufenden Prozedur übernommen werden.
Der Ablauf eines RPC-Aufrufs erfolgt analog in folgenden Schritten (siehe Abbildung unten):

1. Anstoß durch das Client-Programm: die Parameter des Auftrags werden in ein Datenpaket verpackt.
2. Versand des Auftrags und das Auspacken der Parameter im Server-Programm.
3. Durchführung des Auftrags (d.h. der Prozedur) im Server.
4. Verpacken und Versand der Resultate, die zurück an den Client geschickt werden.
5. Auspacken des Resultats im Client und Fortsetzung des normalen Programmablaufs.

Das Ein- und Auspacken der Parameter bzw. des Resultats, die Erzeugung eines vollständigen RPC-Protokollkopfes sowie Absenden und Empfangen der Nachricht wird durch in das Programm eingebundene Bibiliotheksroutinen übernommen.



Simply Network Management Protocol - SNMP (RFC 1157)

Das simply network management protocol erlaubt ein zentrales Netzwerkmanagement für viele Netzwerkkomponenten. Es gibt ***drei Managementbereiche*** in denen dieses Protokoll eingesetzt werden kann:
- beim ***Monitoring*** werden Ereignisse aufgezeichnet und Netzwerkstatistiken gesammelt,
- das ***Controlling*** unterstützt das Ändern von Geräteparametern und -variablen,
- bei der ***Administrierung*** werden Informationen aufgezeichnet, die die Geschichte des Netzwerks beschreiben.

Das Protokoll wurde 1988 definiert. Das Konzept von SNMP basiert im wesentlichen auf einem Netzwerkmanagement-Programm (SNMP Manager), das z.B. auf einer Netzwerkmanagement-Workstation (NMS; network management station) implementiert ist, und SNMP-Agenten in den zu überwachenden Geräten. Der Agent sitzt in einem network node (z.B. Router, Bridge, Terminal Server, Host) und ist für das Sammeln relevanter Managementdaten des Nodes zuständig, die in der management information base (MIB) abgelegt sind. Beispiele solcher Informationen sind Timer, Zähler, Adressen, Protokollparameter. Der Agent reagiert auf SNMP-Anfragen des Managers, durch die MIB-Werte abgefragt oder auch gesetzt werden. Letzteres bewirkt das Steuern des Node. Zusätzlich kann der Agent von sich aus aufgrund bestimmter Ereignisse eine sogenannnte trap message (Trap = Software-Interrupt, Ereignismeldung) senden.

Zur Kommunikation werden die TCP/IP-Protokolle benutzt. SNMP setzt direkt auf der Transportebene auf und verwendet das User Datagramm Protocol (UDP). SNMP unterstützt nur drei Kommandotypen: GET, SET und EVENT (Trap).
Get Request*:* Das Get-Request-Kommando ermöglicht einem Managementsystem die Abfrage einer bestimmten Variablen in der MIB eines Agents.
Get Next*:* Das Get-Next-Kommando ermöglicht die Abfrage von mehreren aufeinander folgenden Variablen einer MIB.
Get Response*:*Das Get-Response-Kommando stellt die Antwort der Agents auf eine durch 'Get Request' oder 'Get Next' erfolgte Anfrage dar.
Set Request*:* Das Set-Request-Kommando ermöglicht die Aenderung einer Variablen innerhalb einer MIB. Auch dieser Befehl wird vom Agent durch ein 'Get Response' quittiert.
Event*:* Während jede Kommunikation immer von der Managementstation aus erfolgt, ermöglicht es das Event-Kommando, einem Agent in bestimmten Situationen (Alarm) eine unaufgeforderte Meldung an die Managementstation zurück zu schicken.

secure SNMP

1992 wurde der IEFT ein Vorschlag für ein sicheres SNMP vorgelegt. Kern ist eine sogenannte party MIB, die festlegt, welche Komponenten in welcher Weise miteinander SNMP-Nachrichten austauschen können. Die Verwendung des DES-Algorithmus (Data Encrypton Standard) wird zwingend vorgeschrieben. Die wichtigsten Elemente von secure SNMP sind aufgegangen in SNMP II.

Simply Managment Protocol - SMP

Das 1988 vorgestellte Netzwerkmanagement-Protokoll SNMP wurde 1993 in einer neueren Version verbessert in Bezug auf Leistungsfähigkeit und Sicherheit. Diese erweiterte Definition hat die Bezeichnung SMP und kann auf Managementplattformen mit SNMP koexistieren. Die

wesentlichen Unterschiede zwischen SNMP und SMP sind folgende: - SMP benutzt nicht nur UDP/IP, sondern auch ein ganzes Spektrum von möglichen Transportprotokollen, darunter Appletalk, IPX und OSI-Protokolle. - In Überarbeitung der Vorschläge des secure SNMP werden Sicherheitsfunktionen angeboten, die bei gleicher Sicherheit eine bessere Leistung als in secure SNMP haben. SMP definiert einige neue Datentypen. Eine SMP-Managementstation kann einer anderen SMP-Managementstation im Rahmen eines verbindungsorientierten Manager-zu-Manager-Kommunikationsdienstes Informationen zukommen lassen und Empfangsbestätigung anfordern. Dazu gibt es eine neue SMP-MIB, die den Informationsaustauch regelt. Diese Kommunikation ermöglicht den Aufbau hierarchischer Managementstrukturen. Ganz besonders wichtig ist die Loslösung von der TCP/IP-Protokollfamilie. Im Unterschied zu secure SNMP macht SMP die Verwendung des DES-Standards zur Option und nicht zwingend. Dies ist auch dadurch begründet, daß die US-Regierung den Export von DES-verschlüsselten source codes verbietet. Ein wichtiges SMP-Ziel ist die Rückwärtskompatibilität zu SNMP: Die gesamten SNMP-MIB-Definitionen können in SMP weiter benutzt werden. Schließlich sehen die SMP-Dokumente Migrationsstrategien und die Koexistenz von SMP- und SNMP-Lösungen auf einer Workstation vor. SMP ist die Grundlage für SNMP II.

Simply Network Management Protocol II - SNMP II

SMP und secure SNMP werden in SNMP II zusammengeführt. Bei den Sicherheitsfunktionen wird durch Authentisierung verhindert, daß Daten verfälscht oder Absender von Daten imitiert werden. Hier ist zur Zeit der sogenannte Algorithmus MD5 im Gespräch. Die Daten selbst werden, falls gewünscht, durch DES verschlüsselt, wodurch eine sehr hohe Abhörsicherheit gewährleistet ist. Letztlich kommt es durch diese Differenzierungsmöglichkeiten zur Bildung von Sicherheitsklassen bei Managern und Agenten, ein auch in anderen Bereichen bewährtes Modell.

Management Information Base - MIB (RFC 1066)

Die MIB ist eine Datenbank, die alle oder viele für ein Management-System relevanten Daten in relationaler oder objektorientierter Form enthält. Die SNMP-MIB ist ein einheitlicher, hierarchisch aufgebauter, protokollunabhängiger Raum für Datenobjekte. Die MIB I enthält mehr als 160 Objekte in acht Gruppen: - system group - interface group - address translation group - IP group - ICMP group - TCP group - UDP group - EGP group (EGP (exterior gateway protocol) - Protokoll zum Austausch von Routerinformationen in IP-Netzwerken) und ist damit völlig auf die Bedürfnisse von Knotenrechnern im Internet abgestimmt Die Systemgruppe beinhaltet Identifikationsmerkmale des Systems und die Interface-Gruppe Anzahl und Art der Schnittstellen. Alle Objekte in der MIB werden einheitlich in ASN.1, der abstract syntax notation one, die ursprünglich für die Definition abstrakter Transfersyntaxen in der Datendarstellungsschicht des OSI-Modells entworfen wurde, formuliert, wodurch eine Normung der abstrakten Darstellungen gegeben ist. Die MIB I reflektiert Protokolle des Internet. Dies ist für die meisten LAN-Anwendungen aber nicht besonders interessant, wenn man z.B. Brücken, Router oder Server im Netz überwachen möchte. Deshab verfolgt man das Konzept der privaten Erweiterungen. Die MIB-Spezifikationen werden laufend erweitert. So arbeitet man heute neben der MIB II, die wesentlich mehr Elemente enthält, an verschiedenen Bereichen wie Host-MIB, Bridge-MIB, Router-MIB usf. Interessant ist auch die RMON-MIB (remote monitoring mit sog. ‘Probes’), die das Sammeln von Stichproben im Netz erlaubt.

HyperText Transfer Protocol - HTTP (RFC 1945)

Das Hypertext Transfer Protocol ist das von den World Wide Web Servern im Internet verwendete Protokoll, um zwischen den Web-Servern und den Web-Clients zu kommunizieren. Es müssen also alle Web-Clients und Web-Server in der Lage sein, HTTP zu sprechen, um Hypertext-Daten senden oder empfangen zu können. Aus diesem Grunde werden Web-Server auch oft HTTP-Server genannt.
Die Dokumente selber sind in der HyperText Markup Language - HTML kodiert, die wiederum eine Anwendung von SGML (Standard Generalized Markup Language) ist. Der Uniform Resource Locator - URL gibt jeweils die Adresse eines Dokumentes im Netzwerk-Server an (bspw. http://www.bkw.ch/boesiger).
Das HTTP geht davon aus, dass als Basis ein verbindungsorientiertes Layer 4 - Protokoll verwendet wird. Im Internet ist das ausschliesslich das Transmisson Control Protocol (TCP). HTTP hat den 'well known' TCP-Port 80. Grundsätzlich würde HTTP auch mit andern verbindungsorientierten Protokollen (bspw. DECnet) arbeiten. In der Praxis hat dies jedoch keine Bedeutung, da die Anwendung hauptsächlich im Internet Verwendung findet.

Eine HTTP-Transaktion zwischen Web-Client und Server besteht aus mehreren Schritten:

- Verbindungsaufbau (Connection)
Der Client baut zum Server eine TCP-Verbindung auf. Als Parameter müssen entweder die IP-Adresse oder der Domain-Name angegeben werden. Wenn nicht anders angegeben, wird der Port 80 (HTTP well known Port) verwendet. Im Normalfall akzeptiert der Server den Verbindungswunsch, ansonsten wird eine Fehlermeldung retourniert.

- Anfrage (Request)
Wenn der TCP-Link steht, sendet der Client dem Server eine Dokumenten-Anfrage. Dies geschiet mit dem Schlüsselwort 'GET' sowie dem Dokumentenname, jedoch ohne Spezifikation des Servers (die TCP-Verbindung auf den Server ist ja schon aufgebaut).

- Antwort (Response)
Die Antwort des Server auf die Anfrage ist eine Hypertext Markup Language (HTML) Meldung. Diese Meldung besteht aus einem ASCII-Zeichen Byte-Strom. Kann das Dokument vom Server nicht lokalisiert werden, sendet er eine Fehlermeldung (in nicht kodierter, von Menschen lesbarer Form im HTML-Syntax) zurück.

- Verbindungsabbau (Disconnection)
Die TCP/IP-Verbindung wird vom Server abgebaut, sobald er dem Client das gewünschte HTML-Dokument transferiert hat. Der Client seinerseits hat jederzeit die Möglichkeit einen Verbindungsabbruch zu forcieren ('Stop'-Ikone beim Browser). In diesem Fall sendet der Server jedoch keine Fehlermeldung zurück!