Einführung in TCP/IP

Andreas Sons
DL8XAS

Hamburger Packet-Radio Gruppe

Dieser Text entstand im Rahmen des Hamburger Amateurfunk Meetings 1996. Er soll einen kurzen Überblick über den Hintergrund der TCP/IP Protokollfamilie geben. Er ersetzt **nicht** die Anleitungen der jeweiligen Programme.

1. Wofür wurde TCP/IP entwickelt?

Um Computer miteinander zu vernetzen gibt es die unterschiedlichsten Technologien und die unterschiedlichsten Protokolle. Im privaten Bereich kann man z.B. ein RS232-Kabel nehmen und mit Kermit Daten übertragen. Wenn dies zu langsam ist greift man zum Ethernet und hat die Auswahl von mehreren Programmen verschiedener Hersteller die sich in der Regel gegenseitig nicht verstehen. Bei Verbindungen über das Telefon dagegen könnte man das Z-Modem-Protokoll verwenden. Dies kann man jetzt beliebig weiterspinnen (ISDN, Weitverkehrsnetze, u.s.w.) aber es bleibt ein Problem: was macht man beim Benutzen unterschiedlicher Netzwerke oder wenn am anderen Ende ganz andere Software verwendet wird?

Um diesem Problem zu begegnen wurde auf Betreiben des US-amerikanischen Department of Defense (auch bekannt als Pentagon) die sogenannte Internet Protocol Suite entwickelt. Ziel der Entwicklung war von der vorhandenen Netztechnik zu abstrahieren und den Anwendungsprogrammen immer die gleichen Schnittstellen zur Verfügung zu stellen. So soll es einem Programm, welches Dateien von einem Rechner auf den anderen überträgt, egal sein ob dies über Funk, Telefon oder einem direkten Kabel geschieht. Ebenso sollte es gleichgültig sein welche Betriebssysteme verwendet werden.

2. IP

Den Mittelpunkt dieser ganzen Familie von Protokollen bildet das Internet Protocol, abgekürzt IP. Dieses nimmt einzelne Pakete von Daten entgegen die sowohl von den übergeordneten Protokollebenen als auch aus dem angeschlossenen Netzwerk stammen können. Ist das Paket für diesen Rechner bestimmt so wird es an das im Paket angegebene Transportprotokoll weitergegeben. Ist es aber für einen anderen Rechner bestimmt so muss das Programm, welches das IP abarbeitet, herausfinden, auf welcher Netzwerkschnittstelle (es könnten mehrere vorhanden sein) dieses Paket weitergeschickt werden muss.

IP erwartet von einem Netzwerk lediglich, daß es Pakete verschicken kann. Die Art und Weise des Transportes ist unwichtig. Weder die Bandbreite interessiert noch ob gelegentlich Pakete verschwinden, in der Reihenfolge vertauscht werden oder Fehler beinhalten. Dies alles sind Probleme um die sich andere Protokolle kümmern müssen.

So kann in Bild 1 z.B. ein Paket bei DL1HDW starten, über ein RS232-Kabel zu DG4HX wandern, von dort aus zu DB0HHS gelangen, dann weiter zu DB0HHT, bei DK3HG schliesslich kommt es in das lokale Ethernet und endet bei DK3HG-1.

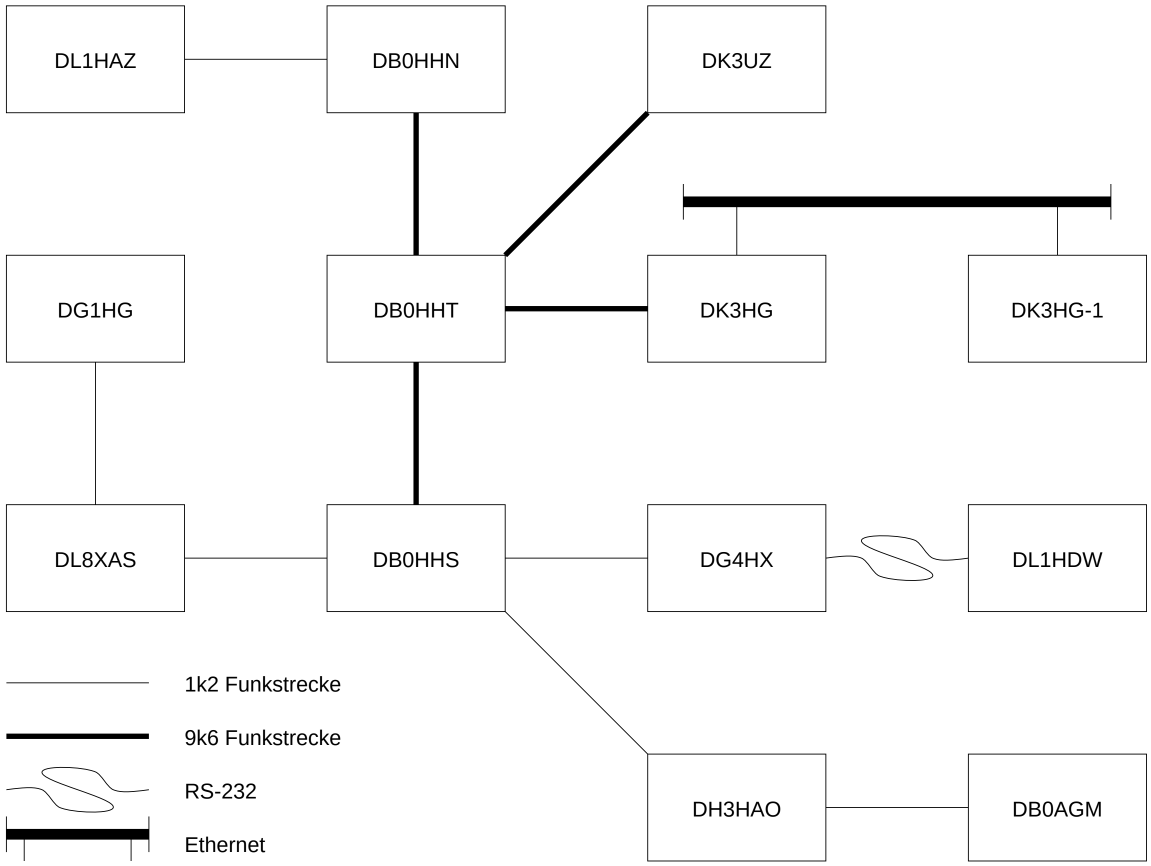

Bild 1 Dies ist ein Beispiel und hat, mit Ausnahme der Rufzeichen, nichts mit der Realität zu tun.

So wie jede normale PR-Station eine Adresse hat, ein Rufzeichen plus SSID, so hat auch jeder TCP/IP-Knoten eine Adresse, allerdings besteht diese aus einer Binärzahl mit 32 Stellen. Diese sogenannte IP-Nummer wird normalerweise als Gruppe von 4 Dezimalzahlen, jeweils getrennt von einem Dezimalpunkt, dargestellt. DB0HHT z.B. hat die IP-Nummer 44.130.0.100. Diese Nummern werden nicht rein willkürlich oder zufällig vergeben sondern nach einem bestimmten Schema. Die 44 weist darauf hin, daß DB0HHT eine Amateurfunkstation ist, eine 26 an dieser Stelle dagegen tragen alle Rechner des MILNET. Die nächste Stelle verweist auf ein Unternetzwerk, die 130 steht für Deutschland. Eine 137 würde bedeuten daß sich der Rechner in den Niederlanden befände. An dritter Stelle steht die Region Deutschlands, die Null heisst Hamburg-Mitte, eine Zwei Hamburg-Süd, eine Sieben Lübeck u.s.w. Mit der letzten Stelle schließlich zählt man die einzelnen Rechner des Unternetzwerks durch.

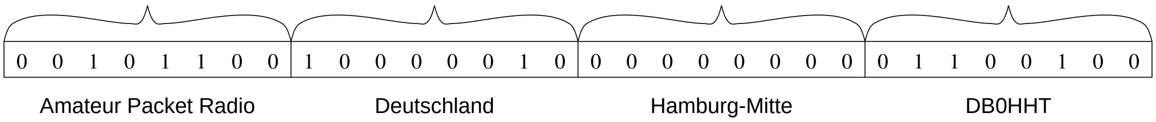

Bild 2

Die Auftrennung in jeweils 8 Bit große Gruppen ist willkürlich, sie kann auch anders erfolgen. Man kann Gruppen oder Unternetze bilden so daß man bei einer geschickten Aufteilung der Adressen auf einzelne Regionen leicht jeden Rechner finden kann. So schickt DB0HHT alle Pakete für Rechner deren Adresse mit 44.130.1 anfängt an DB0HHN da dies IP-Nummern aus der Region Hamburg-Nord sind, Adressen mit 44.130.2 dagegen an DB0HHS.

3. Hostnamen und Domains

Die IP-Nummern sind so leicht zu merken wie Telefonnummern, für Computer ideal, für Menschen weniger. Aus diesem Grund gibt es Hostnamen, im Amateurfunk in der Regel gebildet aus dem Rufzeichen und dem Domainnamen. So sagt der Name DB0HHS.AMPR.ORG aus daß der Rechner DB0HHS heisst und zum AMateur-PacketRadio Netzwerk (AMPR) gehört, welches eine nicht-kommerzielle Vereinigung (ORG) ist. DB0HHS ist der Hostname, AMPR.ORG der Domainname. **Der Hostname ist KEIN Rufzeichen!** Ein Name wie DK3UZ-SVR.AMPR.ORG ist korrekt, kann aber leicht zu Problemen führen. Hier sollte das normale Rufzeichen verwendet werden.

4. TCP

IP ist so zuverläßig wie das drunterliegende Netzwerk. Verliert dieses ein Paket so stellt IP dies nicht fest. Um eine brauchbare Verbindung von einem Rechner auf einen anderen herzustellen gibt es TCP, das Transmission Control Protocol. Dieses nutzt nun die Dienste von IP. Es steckt sozusagen auf einer Seite einer Verbindung Pakete ins Netz und erwartet daß diese in der Regel am anderen Ende auch ankommen. Gehen nun Pakete verloren, kaputt, werden in der Reihenfolge vertauscht oder ähnliches so erkennt TCP dies und wiederholt die Übertragung des Paketes. Das Anwendungsprogramm sagt dem TCP-Programm nur wohin es eine Verbindung haben möchte. Sobald die Verbindung steht erfolgt die Übertragung der Daten. Das Anwendungsprogramm geht davon aus daß, so lange es keine Fehlermeldungen bekommt, die Daten am anderen Ende auch ankommen.

Nun können auf einem Rechner auch mehrere Programme laufen die alle eine Verbindung irgendwohin haben wollen. Um nun die einzelnen Verbindungen den Programmen zuzuordnen verwendet TCP Portnummern. Jeder Endpunkt einer Verbindung hat so eine IP-Nummer, die Kennung des Transportprotokolls (es muss ja nicht TCP sein) und eine Portnummer. Einzelne Dienste haben festgelegte Portnummern. SMTP, ein Protokoll um Post zu verschicken, benutzt z.B. den TCP-Port 25. Wenn ein Programm bei DL1HDW Post an DB0HHT versenden will so sagt es dem TCP-Prozess er möge bitte eine Verbindung an 44.130.0.100:25 aufbauen.

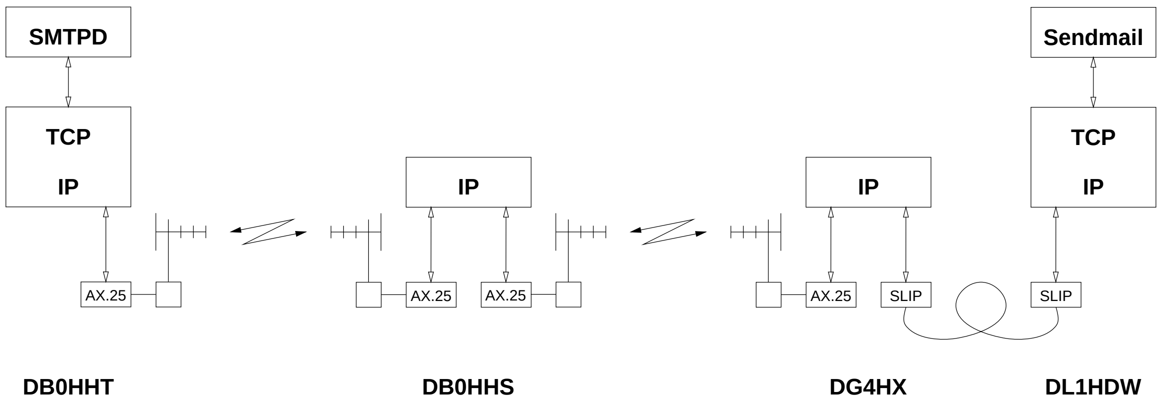

Bild 3

5. Telnet

Mit Hilfe von TCP und IP ist es nun möglich eine Verbindung von einem Programm zu einem anderen aufzubauen. Hierbei ist es unerheblich über wieviele Zwischenstationen oder was für Netzwerkarten die Verbindung läuft. Eine einfache Anwendung währe Kommandos über das Netz an einen anderen Rechner zu schicken. Genau hierfür wurde Telnet entwickelt. Hierbei spielen die Möglichkeiten des Betriebssystems des Rechners, zu dem man eine Verbindung aufbauen möchte, eine große Rolle. Bei Unix oder VMS ist ein sogenanntes Remote Login kein Problem. Anders z.B. bei DOS, dort ist sowas nicht vorgesehen. Bei Software für den Amateurfunk kann man meistens entweder das TCP/IP-Programm selber fernsteuern oder sich mit dem Benutzer am anderen Rechner unterhalten.

6. FTP

Um Dateien von einem Rechner auf einen anderen zu übertragen verwendet man FTP, das File Transfer Protocol. Hiermit kann man Dateien senden, empfangen, löschen, Verzeichnisse auflisten und mehr. In der Übertragungsart ASCII wird der Text auch gleich konvertiert. So haben DOS, MacOS und Unix z.B. unterschiedliche Ansichten darüber welche Zeichen an einem Zeilenende zu stehen haben.

7. SMTP

Eine der interessantesten Möglichkeit von Computernetzen ist das Versenden von Nachrichten. Eine sogenannte Email kann direkt vom Start- zum Zielrechner verschickt werden. Es kann aber auch sein daß diverse Rechner die Nachricht zwischenspeichern müssen weil einzelne Rechner nicht immer erreichbar sind. Wenn DL1HAZ eine Nachricht an DL1HDW verschicken möchte so kann sein Rechner eine TCP-Verbindung zu DB0HHS aufbauen wobei DB0HHN und DB0HHT als IP-Router fungieren (so wie DG4HX in Bild 3) und die Nachricht dort abspeichern. DB0HHS wiederum wartet bis DG4HX erreichbar ist und versucht dann über ihn DL1HDW die Nachricht zuzustellen.

Der Absender muss die Nachricht nur in seinem Rechner eingeben, dieser kümmert sich dann um die korrekte Zustellung. Irgendwann landet sie dann automatisch beim Empfänger. Mit einem entsprechenden Gateway kann man sich auch Post aus dem S&F-Netz der Mailboxen automatisch zustellen lassen.

8. Nameserver, DNS und BIND

Wie bereits erwähnt verwendet IP Nummern zur Adressierung, der Benutzer dagegen Namen. Um nun das Eine in das Andere umzuwandeln kann man eine Tabelle anlegen in der jeweils die IP-Nummer und der Hostname jeder einzelnen Station aufgeführt werden. So eine Tabelle kann aber sehr schnell ins Uferlose wachsen. Um dies zu vermeiden kann man diese Tabelle auf einzelnen Rechnern im Netz vorhalten, alle anderen haben nur die für sie wichtigen Adressen und fragen bei Bedarf nach. Für gerade dieses Nachfragen gibt es das Nameserverprotokoll, auch genannt DNS für Domain Name System. Teilweise begegnet man auch dem BIND, dem Berkeley Internet Name Daemon. Genaugenommen lautet der Protokollname *nameserver*. DNS ist der Name des zugrundeliegenden Konzeptes und BIND nennt sich ein Unixprogramm welches dann als Nameserver arbeitet. Aber gerade hier werden die Begriffe gern und oft durcheinander geworfen.

9. UDP

Nicht für alle Anwendungen lohnt sich der Aufwand eine fehlerfreie Verbindung auf- und dann wieder abzubauen. Beim Nameserverprotokoll wird eine kurze Anfrage gestellt und die entsprechende Antwort erwartet. Würde man hierfür TCP verwenden, so würden haufenweise Pakete übertragen werden die mit dem eigentlich Informationsfluß nichts zu tun hätten. Als Alternative zu TCP gibt es für solche Anwendungen UDP. UDP verläßt sich auf IP und ist entsprechend unzuverläßig. Es fügt den Paketen lediglich eine Prüfsumme hinzu um fehlerhafte Pakete zu erkennen sowie Portnummern wie bei TCP. Geht ein Paket verloren so muß das Anwendungsprogramm dies erkennen und die Aussendung wiederholen.

10. ICMP

Dies ist ein Protokoll um Fehlermeldungen beim Versenden von IP-Paketen zu melden. Es wird benutzt wenn Pakete nicht weitergereicht werden können weil Routen nicht vorhanden sind, Protokolle nicht bekannt sind oder kein Programm vorhanden ist welches das Paket empfangen kann und Vieles mehr. Normalerweise bekommt man dieses Protokoll als Anwender nicht zu sehen. Es gibt aber auch die Möglichkeit das Vorhandensein einzelner Rechner zu testen. Dies wird auch über ICMP abgewickelt.

11. ARP

Wenn DB0HHS ein Paket an 44.130.2.32 verschicken will stellt sich die Frage, welches AX.25-Rufzeichen der Empfänger verwendet. Laut Routingtabelle ist er direkt erreichbar und nach der Hosttabelle heisst er DL0HWL.AMPR.ORG. Nun kann man dies fest vereinbaren, aber da es immer wieder neue Benutzer gibt kann dies ziemlich aufwendig werden. Hier setzt nun ARP, das Address Resolution Protocol an. Es benutzt selber nicht IP, denn genau dieses braucht ja das AX.25-Rufzeichen des Empfängers, sondern steht außerhalb davon. Es stellt auf der Frequenz die Frage *Wer hat die IP-Nummer 44.130.2.32,* und erhält die Antwort *Hier ist DH3HAO, und ich benutze die IP-Nummer 44.130.2.32* Das heisst, daß der Rechner mit dem Namen DL0HWL.AMPR.ORG das Rufzeichen DH3HAO benutzt. Nachrichten für diese Clubstation werden also auch dann zugestellt wenn diese nicht in Betrieb ist. Dies ist **kein** Rufzeichenmißbrauch. Es werden die gültigen AX.25-Rufzeichen verwendet und, wie schon vorher erklärt, hat der Hostnamen nichts mit dem Rufzeichen zu tun. Normalerweise sollte man von solchen Praktiken aber die Finger lassen!!!

12. SLIP, PPP, AX.25 und Ethernet

Wenn nun ein Netzwerktreiber vom IP-Prozess ein Paket erhalten hat und, falls nötig, die Empfängeradresse über ARP erfahren hat, ist das Paket zum Versand fertig. Der AX.25-Treiber wird es in der Regel in ein UI-Paket verpacken und als IP-Paket kennzeichnen. Dies wird erreicht mit einer PID von $CC, entgegen $F0 bei normalen Verbindungen oder $CF bei NetRom. Danach wird es an den TNC übertragen, wobei dieser sich im KISS-Modus befinden muss, da er das Versenden beliebiger Pakete ermöglicht.

Hat man zwei Rechner über ein RS232-Kabel verbunden so wird nur ein minimales Protokoll benötigt um den Anfang und das Ende des Paketes zu kennzeichnen. In der Regel wird dafür SLIP benutzt. Es gibt aber auch PPP, welches sich aber erst lohnt wenn die Verbindung über Telefonmodems läuft. Die Unterschiede zwischen SLIP und PPP würden hier zu weit führen. Wer sich dafür interessiert siehe bitte in der einschlägigen Dokumentation nach.

13. Protocol Stacking

Auf Deutsch auch *Protokolle stapeln.* Wie wohl bereits klar geworden sein dürfte bauen bei TCP/IP mehrere Protokolle aufeinander auf. Die Zusammenhänge sieht man am besten an einer Grafik. Hierbei sind auch Protokolle gezeigt die im Amateurfunk selten oder gar nicht verwendet werden. RIP und EGP sind Routingprotokolle, wobei RIP selten und EGP gar nicht verwendet werden da sie für die Bedürfnisse des Internet ausgelegt sind. Aber auch dort sind sie bereits überholt. Sie dienen hier nur als Beispiel, genau wie NFS, XDR und RPC auch. Das Nameserverprotokoll, in der Grafik als DNS gekennzeichnet, benutzt für kurze Anfragen UDP, wenn aber der Inhalt einer ganzen Domaindatei übertragen werden soll TCP.

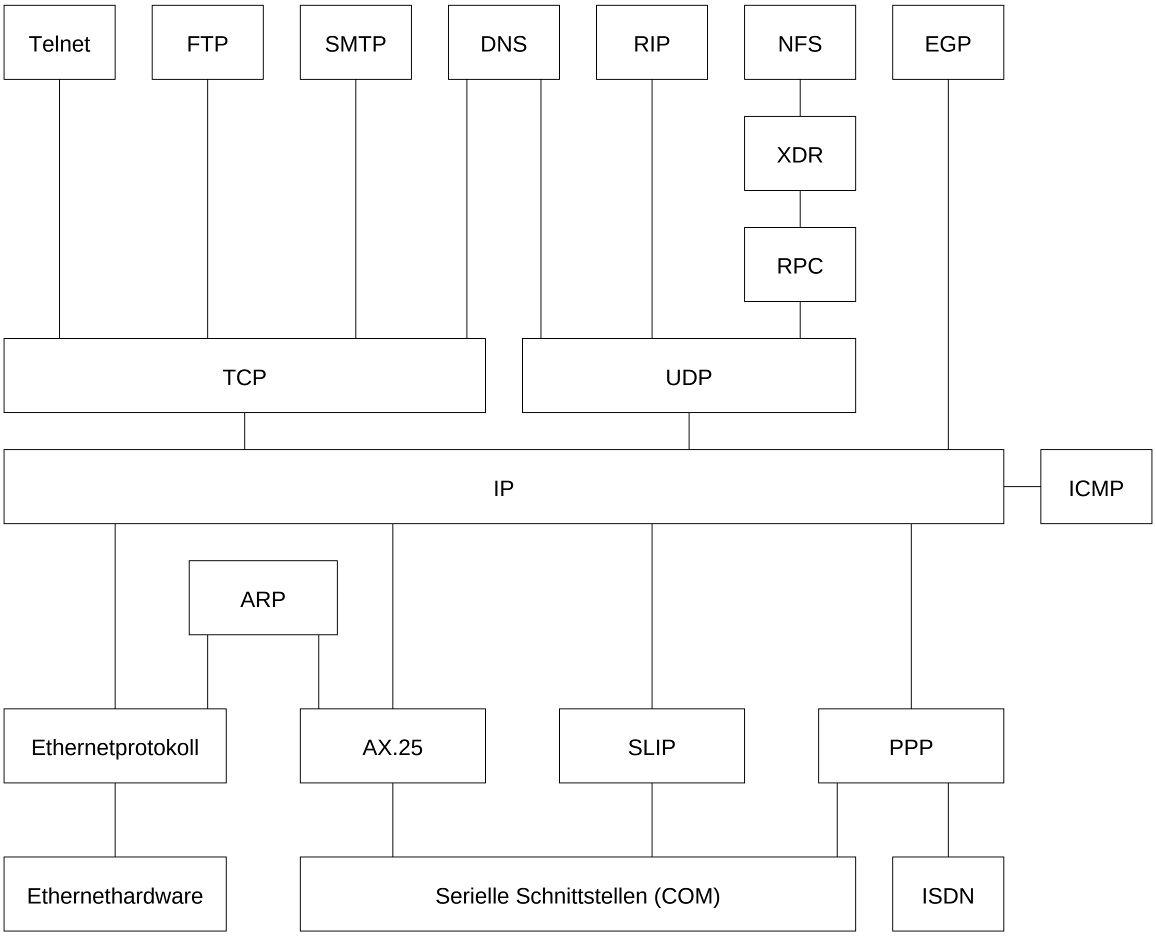

Bild 4 Dies ist ein Beispiel für mögliche Abhängigkeiten. So schließen sich AX.25 und ISDN schon aus rechtlichen Gründen gegenseitig aus und auch EGP und NFS machen im Amateurfunk keinen Sinn.

14. Weiterführende Dokumentation

[1] **Internetworking with TCP/IP, Volume 1** *Douglas Comer,* Prentice-Hall. Band 1 beschreibt die Mechanismen und Protokolle und ist als Einführung gut geeignet. Band 2 und 3 sind für Programmierer gedacht.

[2] **TCP/IP Illustrated, Volume 1: The Protocols** *W. Richard Stevens,* Addison-Wesley. Ähnlich dem Band von Comer, nur noch umfangreicher.

[3] **TCP/IP Network Administration** *Craig Hunt,* O'Reilly & Associates. Konfiguration der Internetprotokolle unter dem Betriebssystem Unix.

[4] **Computer Networks** *Andrew S. Tanenbaum,* Prentice-Hall. Allgemeine Einführung in Netzwerke und ihre Protokolle, basiert stark auf dem OSI-Modell, behandelt aber auch TCP/IP. Auch als deutsche Übersetzung erhältlich.

Die Internetstandarts werden in sogenannten RFCs (Request for Comment) festgelegt. Hierdrin befinden sich die vollständigen Protokollbeschreibungen. Aber nicht alle RFCs behandeln Protokolle, es können auch Vorschläge oder Berichte sein. Hier sind nur die wichtigsten aufgeführt, insgesamt gibt es weit über 1800. Alleine das Verzeichnis der RFCs ist bereits über 300KB groß! Wer sich für ein RFC interessiert siehe im Internet nach oder schicke eine Mail an DL8XAS @ DB0HHT.

768 UDP - User Datagram Protocol

791 IP - Internet Protocol

792 ICMP - Internet Control Message Protocol

793 TCP - Transmission Control Protocol

821 SMTP - Simple Mail Transfer Protocol

826 ARP - (Ethernet) Address Resolution Protocol

854-861
Telnet Protocol specification(s)

959 FTP - File Transfer Protocol