TCP/IP-Grundlagen

Bei diesem Artikel handelt es sich um ein Kapitel aus dem neuen ADIM-Band 71 „Linux“.

August Hörandl

Netzwerk

Mehrere UNIX/Linux Rechner können zu einem Netzwerk verbunden werden. Es können verschiedene Dienste (Post, Datei Transfer, Arbeiten auf anderen Rechnern etc.) genutzt werden. Es sind dabei je nach räumlicher Ausdehnung lokale (LAN) und ausgedehnte Netzwerke (WAN) möglich.

In der UNIX-Welt wird TCP/IP als Standardprotokoll eingesetzt. Linux kann aber auch in heterogenen Netzwerken, d.h. in Netzen mit Rechnern verschiedener Hersteller und Betriebssysteme als Client und Server eingesetzt werden bzw. als Gateway (Schnittstelle) zwischen verschiedenen Protokollen dienen.

Die physische Verbindung kann durch verschiedene Medien erfolgen: serielle und parallele Leitung, Ethernet, Glasfaser, Funk etc. Die verschiedenen Medien sind aber meist, außer durch die Datenübertragungsrate, für den Benutzer nicht unterscheidbar, d.h. er kann die verschiedenen Dienste unabhängig vom Medium nutzen.

Protokolle

Das Transfer Control Protokoll/Internet Protokoll (TCP/IP) wurde ursprünglich als ARPA Netz für militärischen Einsatz entwickelt. Darauf sind einige der zugrundeliegenden Ideen zurückzuführen:

- Es gibt keine zentrale Stelle bzw. keinen zentralen Rechner - alle Rechner sind gleichberechtigt.
- Es gibt keine fixen Verbindungen sondern jedes Datenpaket sucht sich einen Weg durch das Netzwerk.

  Damit ergibt sich eine hohe Verfügbarkeit auch bei Ausfall einiger Rechner bzw. Verbindungen.

In einem TCP/IP Netzwerk wird jedem Rechner (genau: jedem Netzwerkanschluß in jedem Rechner) eine eindeutige Nummer zugeordnet. Diese 32 Bit Adresse wird meist mittels vier durch Punkte getrennter Dezimalzahlen dargestellt. Da jedem Rechner auch ein symbolischer Name zugeordnet ist, kommt ein normaler Benutzer mit diesen Zahlen aber nur sehr selten in Berührung. Diese Zuordnung wird als Domain Name Service (DNS) bezeichnet.

Bei der Kommunikation zweier Rechner wird auch noch eine sogenannte Portnummer zur zusätzlichen Identifikation der Verbindung verwendet. Diese 16 Bit Zahl kennzeichnet einen bestimmten Dienst beim Empfänger des Datenpakets, auf der Seite wird eine freie Nummer dynamisch vergeben. Für die “Standarddienste” existieren vordefinierte Portnummern (well known services). Mit diesen vier Werten (Sender IP Adresse/Portnummer und Empfänger IP Adresse/Portnummer) ist eine Verbindung eindeutig gekennzeichnet.

TCP/IP besteht im wesentlichen aus zwei Protokollen

- User Datagram Protokoll (UDP): verbindungsloses, nicht zuverlässiges Protokoll d.h. Datenpakete können verlorengehen
- Transmission Control Protokoll (TCP): verbindungsorientiertes, zuverlässiges Protokoll

  Beide Protokolle schicken die Daten in sogenannten Paketen. Diese bestehen aus Verwaltungsinformation (Portnummer etc.) und den Nutzdaten.

  Das Internet Protokoll IP sitzt unter diesem Protokoll (Schichtenmodell) und sorgt für die Übertragung und das Routen der in sogenannten IP-Datagramme eingebetteten Pakete zwischen verschiedenen Rechnern. IP weiß dabei nichts von Portnummern, sondern kümmert sich nur um die IP-Adressen. Am Weg durch das Netzwerk werden diese Daten von einem Rechner zum nächsten weitergereicht. Diese als Gateways bzw. Router bezeichneten Rechner sind Teil mehrerer Netzwerke und entscheiden aufgrund der Empfängeradresse wie bzw. wohin das Paket weiter geschickt werden soll.

  Die Information über verfügbare Routen bzw. Gateways wird dabei (nur bei kleinen Netzen) fix definiert oder dynamisch ermittelt, d.h. mittels External Gateway Protocol (EGP), Routing Information Protocol (RIP) oder Border Gateway Protokoll (BGP) zwischen den Routern ausgetauscht. Zusätzlich werden Steuerinformationen bzw. Fehlermeldungen zwischen den Rechnern mit dem Internet Control Message Protokoll (ICMP) ausgetauscht.

  Der Benutzer eines Dienstes sieht aber außer eventuellen Fehlermeldungen (”host not reachable”) nichts von den darunterliegenden Protokollen.

  Obwohl das Internet dezentral organisiert ist, gibt es einige Dienste, die eine “zentrale Stelle” erfordern:

- Jeder Rechner braucht eine eindeutige Nummer
- Die Bezeichnung des Rechners muß ebenfalls eindeutig sein
- Die verwendeten Protokolle müssen weltweit definiert werden.
- Es gibt daher einige Gremien und “Normen”: Jeder kann einen neue Norm vorschlagen (RFC - Request for Comment) - daraus wird dann ein Standard (STD - Beschluß):

NIC	Network Information Center - Vergabe der Internetadressen
ISOC	Internet Society: freiwillige Mitglieder, bestimmt die Richtung der Entwicklung
IAB	Internet Architecture Board: bestimmt welche RFCs zu STDs werden
IETF	Internet Engineering Task Force: Meetings (für alle Internet Benützer), definiert Probleme und Arbeitsgruppen zur Definition neuer RFCs

Hardware

Unter Linux kann TCP/IP über verschiedene Verbindungen übertragen werden:

- Ethernet
- FDDI
- ISDN
- Serielle Verbindungen
- SLIP (Serial Line Internet Protokoll)
- PPP (point to point protokoll) - außer TCP/IP sind auch andere Protokolle möglich; Auch eine dynamische Konfiguration der Verbindungsparameter ist möglich
- X.25 (Paketorientiertes Netz)
- Verbindung über die parallele Schnittstelle (PLIP)
- Verbindung via Funkmodem (AX.25)
- Die verschiedenen Netzwerkkarten, Modems etc. werden über Netzwerkgeräte angesprochen. Die Treiber stellen ein entsprechendes Interface zur Verfügung. Je nach Art der Verbindung ist das entsprechende Gerät nicht immer verfügbar, z.B. nur während der Modemverbindung.

lo	Loopback-Netz
dummy	Pseudoverbindung (wird in Verbindung mit Dial on Demand benötigt)
eth0	erste Ethernet-Karte
eth1	zweite Ethernet-Karte
ppp0	Modem (PPP-Protokoll)
sl0	Modem (SLIP-Protokoll)
plip0	Verbindung über die parallele Schnittstelle
isdn0	ISDN-Karte

Anmerkung

Linux erkennt nur die erste Netzwerkkarte automatisch, alle weiteren müssen entweder beim Booten mit Kernelparametern angegeben werden oder als ladbare Module entsprechend konfiguriert werden.

Netzwerke

Die 32-Bit Adresse ist in zwei Teile geteilt:

- Netzwerk-Teil: bezeichnet ein Netzwerk innerhalb des globalen Internets
- Rechner-Teil: bezeichnet einen einzelnen Rechner innerhalb eines Netzwerks

Je nach Art bzw. Größe des Netzwerks unterscheidet man:

Class	Netz	Rechner	von	bis
A	1 Byte	3 Byte	1.0.0.0	127.0.0.0
B	2 Byte	2 Byte	128.0.0.0	191.255.0.0
C	3 Byte	1 Byte	192.0.0.0	223.255.255.0

Die restlichen Adressen sind für Spezialanwendungen vorgesehen.

Damit sind theoretisch etwa 127 Class A, 16.000 Class B und 2 Millionen Class C Netze verfügbar.

Jeder Netzwerkbetreiber kann eine oder mehrere solche Netzwerknummern beantragen. Die Rechnernummern innerhalb des Netzwerks können dann eigenständig verwaltet werden.

Beispiel

Netzwerk		Rechner	
128	17	75	20

Freie IP Adressbereiche (für interne Netze) - RFC 1918:

10.0.0.0	bis	10.255.255.255	Class A
172.16.0.0	bis	172.31.255.255	Class B
192.168.0.0	bis	192.168.255.255	Class C

Diese Nummern werden nicht vergeben und können für interne Netzwerke verwendet werden. Datenpakete, die eine solche Adresse enthalten, werden im Internet nicht geroutet d.h. sie werden verworfen. Deshalb kann von außen (dem Internet) nicht auf einen Rechner in einem internen Netzwerk zugegriffen werden. Andererseits ist der Zugriff eines Rechners von innen auf das weltweite Internet auch nur mit speziellen Maßnahmen (Proxies bzw. IP Masquerade) möglich.

Teilnetze-Subnetting

Jedes Netzwerk kann weiter in Teilnetze unterteilt werden. Dafür wird der Rechnerteil weiter unterteilt. So kann z.B. ein Class B Netz in 256 Subnetze unterteilt werden. Die Unterteilung muß dabei nicht an den Byte-Grenzen der Adresse erfolgen, sondern es sind auch andere Unterteilungen möglich. Man kann z.B. ein Class C Netz (256 Adressen) weiter unterteilen bzw. man erhält von einem Provider nur ein Subnetz mit z.B. 8 Adressen. Dabei ist aber zu beachten, daß in jedem Netzwerk die niedrigste und höchste Adresse als Netzmaske bzw. als Broadcastadresse nicht für einen Rechner verwendbar sind, d.h. man verliert durch diese Teilnetze Adressen.

Die Unterteilung erfolgt durch Angabe einer Maske (subnetmask): alle Bits die übereinstimmen müssen, werden auf 1 gesetzt, der Teil der Rechnernummer auf 0.

Beispiel

Netzwerk		Subnetz	Rechner
128	17	75	20

Netmask: 255.255.255.0

Broadcast Adresse (alle Rechner im Netzwerk): 128.17.75.255

Diese Unterteilung ist aber nur ''intern''; nach ''außen'' erscheint weiterhin ein Netzwerk. Damit können aber zwei Subnetze oder zwei Rechner eines Netzwerks nicht an ''verschiedenen'' Punkten des Internets eingebunden werden: das Routen geschieht immer auf Grund von Netzwerken und nicht für Einzelrechner oder Teilnetze.

Anmerkung

Auch mit dieser Einschränkung muß ein Router theoretisch die ''Richtung'' zu mehreren Millionen Netzwerken kennen bzw. speichern.

Das in Vorbereitung befindliche, neue Internet protokoll (IPv6, IPNG) soll das Routen vereinfachen und zusätzliche Adressen zur Verfügung stellen (64 statt 32 Bit Adressen).

Die IP Adressen können in einem Netz fix vergeben werden oder erst bei Bedarf dynamisch zugewiesen werden. Mit einem Bootp oder DHCP Server kann diese Vergabe und Verwaltung zentral erfolgen. Auch bei fixen IP Adressen kann dies von Vorteil sein, da alle Änderungen zentral durchgeführt werden können. Eine dynamische Vergabe ist bei Rechnern, die ''weltweit'' erreichbar sein sollen, problematisch, bei internen Netzen aber durchaus sinnvoll. Die Fehlersuche ist aber etwas schwieriger.

Domain Name Service - DNS

Einführung

Jeder Rechner hat eine IP Adresse: eine 32 Bit Zahl, für die Darstellung meist in 4 Byte unterteilt und dezimal angegeben. Zur Vereinfachung werden aber Namen vergeben. Die Umsetzung erfolgt durch sogenannte Nameserver. Unter Linux heißt dieses Programm named.

Man kann einen Nummernbereich beantragen (eine Domain) und die darin enthaltenen Netze bzw. Subnetze und Rechnernamen selbst verwalten. An die übergeordnete Stelle wird nur die Adresse des Nameservers bekanntgegeben.

Die Zuordnung Name - Nummer ist nicht eindeutig:

- Ein Rechner kann mehrere verschiedene Nummern haben:
- eine in jedem Subnetz, dem er angehört, d.h. eine je Netzwerk.
- Ein Rechner kann verschiedene Namen haben:
- Virtuelle Server: ein Rechner stellt je nach verwendeten Namen verschiedene Informationen zur Verfügung.
- In einem kleinen Netzwerk gibt es einen Rechner als Mail, WWW und FTP Server; entsprechend hat der Server die Namen *www*, *ftp* und *mail*. Bei stärkerer Belastung wird Hardware ergänzt und die verschiedenen Server und die Namen werden auf mehrere Rechner aufgeteilt - für die Benutzer ergibt sich kein Unterschied.

Beispiel

193.170.162.14 *elina.htlw1.ac.at*

Die Nummer ist von links nach rechts zu lesen: die HTL W1 hat die Adressen 193.170.162.0 bis 193.170.162.255 (Class C Adresse - letztes Byte als Rechnernummer); dieser Bereich könnte noch unterteilt werden. Jeder Rechner bekommt (mindestens) eine Nummer und (mindestens) einen Namen (*elina*).

Der "weltweite" Name lautet *elina.htlw1.ac.at* - der Name ist von rechts nach links zu lesen:

at	Österreich (Top Level Domain)
ac	akademisches Netz
htlw1	HTL Wien 1
elina	Rechnername

Wichtige Domains

USA bzw. Weltweit

.com	kommerzielle Unternehmen
.edu	Wissenschaft (Universitäten)
.gov	Regierung
.org	(gemeinnützige) Organisationen

Länderbezeichnung

.at	Österreich
.ch	Schweiz
.de	Deutschland
.uk	Großbritannien

Innerhalb eines Landes wird meist weiter unterteilt (ähnlich *com, edu, org*...)

z.B.: *.co.at, .ac.at, .or.at*, ...

Umsetzung Rechnername in IP Adresse

Beim Domain Name Service (DNS) handelt es sich um eine riesige Datenbank. Mit ihrer Hilfe ist die Umwandlung eines Rechnernamens in eine eindeutige Rechnernummer möglich. Da kein Rechner alle Namen wissen kann, ist diese Datenbank auf viele Rechner verteilt. Im Allgemeinen verwaltet dabei jedes Netzwerk (Domain) seine eigenen Namen und einen übergeordneten Server kennt den Nameserver für dieses Netzwerk. Damit ergibt sich auch die Notwendigkeit von obersten oder Root Name Servern. Zur Erhöhung der Verfügbarkeit gibt es neben dem ''Primary Nameserver'' meist einen oder mehrere ''Secondary Nameserver''. Ein Nameserver muß nicht nur die Umsetzung eines Namens in eine IP Adresse sondern auch das sogenannte Reverse Lookup, d.h. Nummer in Name, beherrschen.

Beispiel

Irgendwo will jemand mit seinem Browser eine Seite am Rechner elina.htlw1.ac.at lesen. Dazu muß der Name mittels DNS in eine IP Adresse umgewandelt werden.

Eine Anfrage wird an den lokalen Nameserver geschickt. Da dieser die Umsetzung selbst nicht vornehmen kann, muß einer der Root Name Server abgefragt werden. Dieser weiß die Adresse nicht, er vermittelt aber zum Nameserver für den Bereich .at weiter. Dieser Server teilt auf Anfrage den Nameserver für .ac.at mit und schließlich wird er zum Nameserver des Netzes htlw1.ac.at weitergeschickt. Jetzt kann der Name endlich in eine Nummer umgewandelt werden und der lokale Nameserver schickt das Ergebnis an den Client.

In der Praxis können einige dieser Abfragen entfallen, da Nameserver alle Antworten speichern und damit viele Anfragen aus dem Speicher beantworten können. Jedem Eintrag in der Datenbank ist eine Lebensdauer zugeordnet, erst nach Ablauf muß eine Auskunft erneut eingeholt werden.

Aus diesem Grund ist ein lokaler Nameserver auch dann sinnvoll, wenn man keine eigenen Domain zu verwalten hat. Ein solcher cache-only Server beschleunigt die DNS Zugriffe in einem LAN.

Anmerkung

Die Unterteilung in Netzwerke (Class A - Class C) und die Einteilung in Domains hat nur bedingt etwas mit einander zu tun. Die Domain .at besteht (wahrscheinlich) aus vielen Class A, Class B und Class C Netzen.

TCP/IP Dienste

Start der Server

Allgemeines

Die verschiedenen Dienste werden durch Server oder Dämone zur Verfügung gestellt. Diese Programme öffnen ein Port, d.h. ein Ende einer Netzwerkverbindung, und warten nach dem Start bis ein Client eine Verbindung aufnimmt. Meist wird dann ein Kindprozeß gestartet und dieser behandelt die Anfrage. Dieses Verfahren hat den Nachteil, daß für alle möglichen Services bereits ein Server gestartet sein muß und dadurch dauernd Systemressourcen belegt werden. Andererseits ist die Antwortzeit und die dynamische Systembelastung klein, da nur wenige Prozesse gestartet und gestoppt werden müssen. Das geht so weit, daß einige Server (z.B. der Apache WWW Server) einige Kindprozesse auf Vorrat erzeugen.

Die Alternative besteht darin, den Server erst bei Bedarf zu starten. Nur die Server, die dauernd bzw. oft gebraucht werden oder sehr viel Zeit beim Starten brauchen, werden schon beim Systemstart aktiviert.

Inetd

Fast alle UNIX Installationen haben einen Superserver, der auf allen Ports lauscht und bei Bedarf den entsprechenden Serverprozeß startet.

Dieser Superserver heißt inetd. Er wird beim Booten gestartet und holt sich die Liste der zu betreuenden Dienste aus der Datei /etc/inetd.conf. Damit können die zur Verfügung gestellten Services ein- und ausgeschaltet werden.

Beispiel

```
# service type protocol wait user
server cmdline
telnet stream tcp nowait root
/usr/sbin/telnet telnetd -b/etc/issue
```

Die Angabe der Portadresse erfolgt über einen Namen. Die Zuordnung zwischen diesem symbolischen Bezeichnung und der Portnummer erfolgt für die "well known services" in der Datei `/etc/services`:

```
# service port/protocol alias
echo          7/tcp
echo          7/udp
telnet       23/tcp
printer     515/tcp    spooler
...
```

Die Umsetzung der hierverwendeten Protokollbezeichnungen in Nummern erfolgt in der Datei `/etc/protocols`

```
ip   1 IP
tcp  6 TCP
...
```

RPC

Ein weiterer wichtiger Mechanismus für die Client-Server Kommunikation sind die "remote procedure calls" (RPC). Diese bilden die Basis für verschiedene andere Dienste z.B. NFS und NIS.

Der Client ruft dabei eine Funktion (Unterprogramm) auf. Die Parameter des Aufrufs werden an den Server übermittelt und dieser führt die Funktion aus. Die Ergebnisse werden wieder zurück an den Client geschickt. Für die Übertragung werden alle Daten in eine rechnerunabhängige Darstellung umgewandelt. Damit können auch verschiedene Rechner mit unterschiedlicher interner Zahlendarstellung miteinander kommunizieren.

Dieses Service wird durch Start des Portmapper Dämons rpc.portmap zur Verfügung gestellt. In der Datei `/etc/rpc` findet sich die Zuordnung der verschiedenen Funktionsnamen auf Nummern:

```
# /etc/rpc
portmapper 100000  portmap sunrpc
nfs        100003  nfsprog
...
```

Konfiguration

Die wichtigsten Dateien

In den HOWTO Dokumenten finden sich Erläuterungen und viele Beispiele zur Konfiguration und Betrieb eines Netzwerkes.

/etc/hosts	Zuordnung IP Adresse - Hostname
/etc/networks	Zuordnung Netzwerknamen - IP Nummern
/etc/host.conf	wie werden Namen in Nummern umgesetzt
/etc/resolv.conf	Angabe der Domain des Rechners (wird gegebenfalls an den Namen angehängt) und des Nameservers
/etc/named.boot	Konfiguration des Nameservers
/etc/HOSTNAME	enthält den vollständigen Rechnernamen (mit Domain!)

Beispiel

Nameserver `/etc/named.boot`:

```
directory /var/named
cache . root.cache
forwarders ns.provider.at
slave
```

Auflösen der Namen `/etc/resolv.conf`:

```
search htlw1.ac.at
nameserver 127.0.0.1 10.1.1.1
```

Reihenfolge bei der Suche `/etc/host.conf`

(zuerst `/etc/hosts`, dann Nameserver):

```
order hosts bind
multi on
```

In der nächsten Ausgabe folgt ein Beitrag über das Editieren unter Linux.

August Hörandl hoerandl@elina.htlw1.ac.at