UNIX Netzwerk-Tools

Johann Schlamp

29.11.2005

Inhaltsverzeichnis

1 Netzwerk-Aufbau

Der Begriff **Netzwerk** wird allgemein definiert als ein Verbund zweier oder mehrerer Geräteeinheiten. Die Verbindung unterschiedlicher *Netzwerk-Segmente* erfolgt über Koppelgeräte (z.B. Hubs, Switches, Bridges, Router u.s.w). Es ist kein willkürliches Senden von beliebigen Daten erlaubt, sondern nur der Versand von *Paketen* fest definierter Länge.

1.1 WAN – LAN

1.1.1 LAN (Local Area Network)

Nach der Definition der *International Standard Organisation (ISO)* sind LANs lokale Netzwerke, die auf das Gelände des Benutzers beschränkt sind und einzig und allein in dessen *rechtlichem Entscheidungsbereich* liegen.

1.1.2 WAN (Wide Area Network)

WANs dagegen sind Weitverkehrsnetze, bei denen einzelne Teile desselben *logischen Netzwerks* über große Entfernungen miteinander verbunden sind. Dazu müssen oft öffentliche Netze benutzt werden, die von sogenannten *Providern* zur Verfügung gestellt werden. Der Benutzer hat dadurch nicht mehr die alleinige Kontrolle über das Netzwerk, einen großen Teil der Verantwortung trägt hier der Provider.

1.2 Topologie

Ursprünglich beschreibt die Topologie die *Lage und Anordnung* von mathematischen Gebilden im Raum. In der Informatik wird der Begriff aber vor allem zur Beschreibung von **Vernetzungsarten** verwendet.
Die zugrundeliegende Struktur eines Netzwerks ist unter anderem ausschlaggebend für die

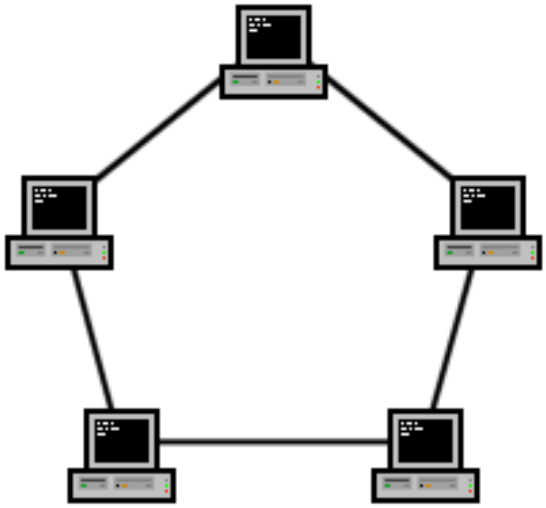
- Ausfallsicherheit
- Performance
- Abhörsicherheit und
- Investitionskosten.

Man unterscheidet zwischen **physischer** und **logischer** Topologie. Bei der physischen Topologie interessiert man sich, wie die Geräte tatsächlich untereinander verbunden sind, wogegen die logische Topologie das Verhalten der Geräte, also die tatsächliche Nutzung des gegebenen Aufbaus beschreibt. Die wichtigsten Beispiele für (physische) Topologien werden im Folgenden kurz vorgestellt.

1.2.1 Bus-Netz

Zentrales Merkmal von Bus-Netzen ist die gemeinsame Datenleitung, die sich alle Geräte teilen müssen. Vorteil davon sind geringe Kabelmengen und ein relativ kleiner Aufwand bei der Planung und Erweiterung des Netzwerks.
Es gibt jedoch auch etliche Nachteile. Fällt der Datenbus aus, so ist das gesamte Netzwerk lahmgelegt. Außerdem führt die Kommunikation über nur eine einzige Leitung zu häufigen Paketkollisionen und damit zu einer niedrigeren Geschwindigkeit bzw. einer kleineren maximalen Auslastung als bei anderen Topologien. Ein weiterer großer Nachteil ist die Anfälligkeit gegenüber Lauschangriffen, da Daten über den Bus an alle Teilnehmer und somit möglicherweise auch an unerwünschte Gäste gesendet werden.

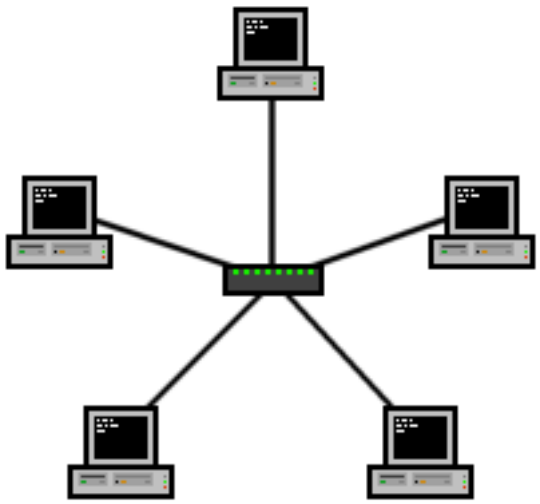
1.2.2 Ring-Netz



Ring-Netze sind im Gegensatz zu Bus-Netzen viel sicherer vor Angriffen, da die Kommunikation deterministisch ist. Daten werden nicht an das gesamte Netzwerk übermittelt, sondern durchlaufen den Ring nur bis zum vorgesehenen Empfänger. Desweiteren ist eine

Senderichtung festgelegt, wodurch Kollisionen praktisch ausgeschlossen sind. Außerdem arbeiten alle Stationen als Verstärker, d.h. es sind theoretisch beliebig große Netzwerke ohne zusätzliche Geräte möglich.
Großer Nachteil ist allerdings, dass der Ausfall eines einzigen Endgerätes das komplette Netz lahmlegt.

1.2.3 Stern-Netz



Bei dieser Art der Vernetzung werden alle Geräte mit einem zentralen Verteiler verbunden. Dabei kann zwischen zwei Teilnehmern eine exklusive Verbindung aufgebaut werden, während die restliche Kommunikation unbeeinflusst bleibt. Stern-Netze bieten dadurch sehr hohe Übertragungsraten. Zusätzlich hat der Ausfall eines Endgerätes keine Auswirkungen und das Netzwerk ist leicht erweiterbar.
Dafür sind aber offensichtlich große Kabelmengen möglich, was diese Topologie sehr kostenträchtig macht.

Es sind noch viele weitere Topologien wie **Baum-, Zellen-** und **vermaschte Netze** im Einsatz, die aber meistens aus Kombinationen der obigen Strukturen aufgebaut sind.

1.3 Identifizierung von Geräten

Um eine sinnvolle Kommunikation in einem Netzwerk zu gewährleisten, muss jeder Teilnehmer adressiert werden können. Dies ist sowohl in privaten bzw. kleineren Netzwerken als auch **weltweit eindeutig** nötig.
Dazu werden die sogenannten *MAC-Adressen* und *IP-Adressen* vergeben.

1.3.1 MAC-Adressen

MAC steht für *Media Acces Control* und ist eine Schicht des **OSI-Modells**. Dieses legt den theoretischen Grundstein zum Betrieb von Netzwerken, indem die Kommunikation in verschiedene Ebenen unterteilt und dort separat betrachtet wird.
Um die weltweit eindeutige Identifizierung sicherzustellen, sind MAC-Adressen fest in die Hardware „eingebrannt“. Der Aufbau aus **48 Bit** ermöglicht 2^{48} (mehr als 100 Billionen) unterschiedliche Adressen.
Zur besseren Übersicht werden MAC-Adressen meist als acht Hexzahlen-Blöcke (12 Ziffern bzw. Buchstaben) geschrieben.

Hersteller von netzwerkfähigen Geräten müssen darauf achten, dass jedes Gerät eine einzigartige Adresse besitzt. Um Konflikte zwischen verschiedenen Herstellern zu vermeiden, wird die Vergabe von einer zentralen Stelle, dem *Institute of Electrical and Electronical Engineers*, kurz *IEEE*, verwaltet.

Die ersten **24 Bit** (3 Hex-Blöcke) der MAC-Adresse identifizieren daher **Hersteller** und oft auch den **Typ** der Netzwerkkarte. Ein paar Beispiele:

`00-05-8B-xx-xx-xx`	Compaq
`02-05-09-xx-xx-xx`	Hewlett-Packard
`00-07-E9-xx-xx-xx`	Intel
`08-05-20-xx-xx-xx`	Sun
`09-04-75-xx-xx-xx`	3Com

Die komplette Liste aller vergebenen MAC-Adressen ist von der IEEE Homepage unter http://standards.ieee.org/regauth/oui/oui.txt abrufbar.

1.3.2 IP-Adressen

Zusätzlich zur bisherigen allgemeinen Adressierung ist eine **logische Untergliederung** von Netzwerken in unterschiedliche Teilbereiche sehr sinnvoll. Um dies (und eine Vereinfachung von den „kryptischen“MAC-Adressen) zu erreichen, wurden sogenannte **IP-Adressen** eingeführt. IP steht für *Internet Protocol*, das die Kommunikation im OSI-Schichtenmodell auf einer höheren Ebene implementiert.
Um Konflikte zu vermeiden, ist weiterhin eine eindeutige Identifizierung von geräten im Netzwerk wichtig. Dazu werden IP-Adressen über das sogenannte **Adress Resolution Protocol (ARP)** eindeutig an MAC-Adressen gebunden. Soll eine IP-Adresse in eine Hardware-Adresse aufgelöst werden, wird ein *arp-request* an alle Teilnehmer des Netzes gesendet. Jedes Gerät, das dazu in der Lage ist, muss antworten. Um den daraus resultierenden Verkehr gering zu halten, werden die IP/MAC-Zuweisungen nicht dauerhaft gespeichert, sondern nur für wenige Minuten im *arp-cache* gehalten.

IP-Adressen bestehen aus **32 Bit**. Die Anzahl unterschiedlicher Adressen liegt dadurch

in der Größenordnung von einer Milliarde. Beachtet man, dass in naher Zukunft alle elektronischen Geräte miteinander vernetzt werden sollen, erscheint dieser Vorrat relativ klein. Abhilfe schafft das neuere **Internet Protocol v.6**, das mit **128 Bit** langen Adressen theoretisch jedem Atom der Erde eine eindeutige Adresse liefern könnte.
Wichtiges Merkmal der IP-Adressen ist die Unterteilung in **Netzwerk-** und **Geräteteil**. Die Trennung erfolgt mit der sogenannten **Subnetz-Maske**. Das Prinzip dabei sind weitere 32 Bit, aufgeteilt in zwei Blöcke. Der erste besteht nur aus gesetzten Bits (Einsen) und definiert damit die Länge der Netzadresse, der zweite Block ist nur mit Nullen gefüllt, die die Länge der Geräteadresse angeben.
Die IP-Adresse **192.168.0.1** mit der Netzmaske **255.255.255.0** beispielsweise sieht in binärer Schreibweise folgendermaßen aus:

`11000000.10101000.00000000.00000001`	IP-Adresse
`11111111.11111111.11111111.00000000`	Netz-Maske

Man sieht leicht, dass die Netzadresse den ersten 24 Bit, also **192.168.0** entspricht. Außerdem ergeben sich daraus für den Geräteteil noch 8 freie Bits, mit denen maximal $2^8 = 256$ verschiedene Geräte adressiert werden können.

In privaten Netzwerken werden IP-Adressen entweder **statisch**, d.h. manuell durch den Benutzer oder **dynamisch** über das *Dynamic Host Configuration Protocol (DHCP)* vergeben.
Größeren Einrichtungen wie Firmen, Universitäten, u.s.w. werden von einer zentralen Verwaltungsstelle feste Adressbereiche zugewiesen.

1.4 Tools

1.4.1 arp

arp [-H type] -a [hostname]
arp [-H type] -d [hostname]
arp [-H type] -s hostname hw-addr

Das Tool *arp* gibt mit dem Parameter *-a* den Inhalt des arp-caches (auf Wunsch nur für einen bestimmten Hardware-Typ oder Hostname) aus. Außerdem ist es möglich, Einträge zu löschen (*-d*) oder zu erzeugen (*-s*).

1.4.2 ifconfig

ifconfig [interface]
ifconfig interface up |down |netmask | [address]

Mit *ifconfig* kann man sich alle verfügbaren Informationen wie MAC-Adresse, IP-Adresse,

Netzmaske u.s.w. von allen (oder nur einem bestimmten) Netzwerk-Interface anzeigen lassen.
Zusätzlich können Interfaces ein- und ausgeschaltet (*interface up | down*), sowie die Netzmaske und IP-Adresse verändert werden.

1.5 Beispiel Ethernet

Ethernet ist eine weit verbreitete **Vernetzungstechnik** für LANs, die durch den Standard *IEEE 802.3* definiert wird. Dort werden Kabeltypen (RJ45, Glasfaser, ...) und Paketformate festgelegt, die Protokolle der Hardware-Schicht spezifiziert und eine Bus-Topologie gefordert. Ethernet arbeitet also *logisch* immer als Bus-Netz, obwohl die wirkliche Verkabelung oft eine andere ist.

2 Netzwerk-Betrieb

Um Netzwerke effizient betreiben zu können, muss weiter von der Hardware abstrahiert werden. Der Benutzer soll sich weder darum kümmern müssen, auf welche Art seine Daten und letztendlich seine „Bits und Bytes“ übertragen werden, noch wie er Geräte über MAC- oder IP-Adressen ansprechen muss.
Es sind zum einen **aussagekräftige Namen** zur Identifizierung im Netzwerk und zum anderen viele **weitere Protokolle**, die auf dem Internet-Protokoll aufbauen, nötig. Die inneren Abläufe sollen weitestgehend verborgen bleiben, um ein benutzerfreundliches Arbeiten mit Netzwerken zu erlauben.

2.1 Protokolle

Die Aufgabe von Protokollen ist allgemein das **verlässliche Übertragen** von Daten. Es gibt viele Unterscheidungsmerkmale:

- *Unicast – Multicast:*
  Gibt es nur einen oder mehrere Empfänger?
- *Simplex – Halb-Duplex – Voll-Duplex:*
  Findet der Verkehr nur in eine Richtung, abwechselnd oder gleichzeitig in mehrere Richtungen statt?
- *Peer-to-Peer – Client-Server:*
  Sind alle Teilnehmer gleichberechtigt oder gibt es unterschiedliche Aufgaben?
- *synchrone – asynchrone Kommunikation:*
  Müssen alle Teilnehmer gleichzeitig online sein, oder kann die Kommunikation versetzt erfolgen?
- *paketorientiert – streamorientiert:*
  Werden einzelne Pakete in beliebiger Reihenfolge versandt oder ist ein kontinuierlicher Strom nötig?

Beispiele für einige Protokolle aus verschiedenen Schichten sind

Physik:	**Ethernet, Token Ring**
Vermittlung:	**IP, ICMP, IPX**
Transport:	**TCP, UDP**
Anwendung:	**HTTP, FTP, SMTP, SSH**

Die folgenden Programme sind in der Vermittlungsschicht angesiedelt und benutzen das *ICMP-Protokoll*.

2.1.1 ping

ping [-b] [-c count] destination

Mit *ping* wird eine Serie von Paketen an die angegebene Zieladresse gesandt. Das zugrundeliegende Protokoll verlangt, mit gleichwertigen Paketen zu antworten. Die Option *-c* sendet nur eine bestimmte Zahl (*count*) Pakete, mit *-b* wird ein sogenannter **Broadcast**, also eine Anfrage an alle erreichbaren Teilnehmer gesendet.
Der große Nutzen dieses Tools liegt im schnellen und unkomplizierten Testen der Netzwerverbindung.

2.1.2 traceroute

traceroute [-m max-hops] destination

traceroute ist *ping* sehr ähnlich. Hier wird aber jedes einzelne Gerät bis zur angegebenen Adresse (mit *-m* über höchstens *max-hops* Stationen) ausgegeben. Man kann damit also den Weg eines Paketes von der Quelle bis zur Zieladresse über Server des Providers o.ä. verfolgen.

Nachfolgend werden einige Programme für verschiedene Protokolle der *Anwendungsschicht* aufgeführt.

2.1.3 ftp

ftp [-p] [hostname [port]]

FTP (File Transfer Protocol) ist ein Protokoll zum Übertragen von Daten. Dazu werden mehrere Verbindungen zwischen Client und Server aufgebaut. Nach deren Anzahl unterscheidet man zwischen **Active Mode** und **Passive Mode**.
Das Tool *ftp* benutzt standardmäßig den Active Mode, mit *-p* kann in den Passive Mode gewechselt werden. Außerdem kann die Adresse und ein eigener Port für den Datenaustausch angegeben werden.

Beim Aufruf ohne Parameter startet eine Kommando-Umgebung, in der mit der Eingabe *?* eine Übersicht über alle Befehle erscheint.

2.1.4 ssh

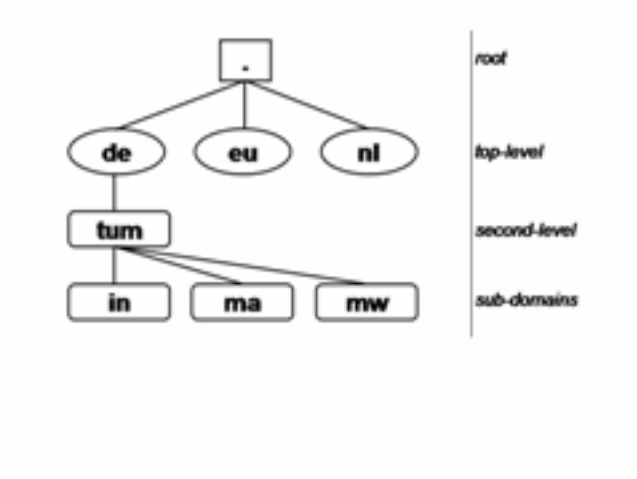
ssh [hostname | user@hostname]

Ein anderes wichtiges Protokoll ist **ssh (Secure Shell)**. Es dient zum Einloggen auf entfernten Computern über eine *authentifizierte* und *verschlüsselte* Verbindung. Dabei gibt es die Möglichkeit zur Ausführung von Programmen, Übertragung von Daten und Weiterleitung von *X11-Sitzungen*. Man kann damit arbeiten, als würde man direkt an dem entsprechenden Computer sitzen.
ssh ist sehr einfach zu benutzen. Als Parameter wird nur die Zieladresse (bzw. direkt der Loginname) übergeben. Es erscheint der Login-Prompt auf dem Zielrechner.

2.2 Namensauflösung

Um die Benutzerfreundlichkeit zu erhöhen, müssen die unhandlichen Zahlen der MAC- bzw. IP-Adressen ersetzt werden durch **aussagekräftige** und einfach zu merkende **Namen**. Dabei ist natürlich wieder auf eine **(weltweit) eindeutige Zuweisung** zu achten. Das heutige Internet wäre ohne diese Erweiterung nicht denkbar.

Für die Realisierung ist das **DNS (Domain Name System)** zuständig. Um Namen möglichst effizient in IP-Adressen aufzulösen, ist es als *verteilte Datenbank* konzipiert.
Der Namensraum ergibt sich durch die Verkettung sogenannter **Labels**, die jeweils aus maximal 63 Zeichen bestehen können und durch Punkte getrennt werden. Die Hierarchie dabei ist von rechts nach links. Aus diesen Vorgaben ergibt sich eine **Baumstruktur**:



Eine (Internet-) Adresse wird meist *rekursiv* über mehrere **Nameserver** aufgelöst. Die Anfrage wird zuerst an den *rootserver* geschickt, der alle Nameserver der nächsten beiden Labels, der **Toplevel-** und **Secondlevel-Domains**, kennt. Dementsprechend wird

der Request rekursiv weitergeleitet, bis der Name in eine IP-Adresse aufgelöst ist.
Die Inverssuche (*reverse lookup*) ist ebenfalls möglich.

Abschließend folgen einige Tools, mit denen DNS-Einträge abgefragt werden können.

2.2.1 host

host [-i adress | hostname]

Mit *host* kann zu jedem Hostname die zugehörige IP-Adresse oder mit *-i* zu jeder IP-Adresse der entsprechende Hostname bestimmt werden.

2.2.2 dig

dig @hostname [-x]

dig macht prinzipiell dasselbe wie *host*, nur werden hier weitaus mehr Informationen (z.B. alle *Sub-Domains*) mitgeliefert. Mit *-x* ist die Inverssuche möglich.

2.2.3 nslookup

nslookup [norecurse] [name] [nameserver]

Das etwas ältere Tool *nslookup* ermöglicht ebenfalls eine DNS-Abfrage, wobei ein spezieller Nameserver oder ein iteratives Verfahren zur Suche verwendet werden kann.

Selbstverständlich gibt es viel mehr zu jedem einzelnen Teilgebiet zu sagen, und sicherlich waren die hier vorgestellten Tools nur ein kleiner Ausschnitt – aber für den täglichen Bedarf beim Umgang mit Netzwerken sind sie oft schon ausreichend.