Netzwerk

Linux-Kurs der Unix-AG

Andreas Teuchert

18./19. Juli 2012

Netzwerk-Protokolle

- legen fest, wie Daten zur Übertragung verpackt werden
- unterteilt in verschiedene Schichten:
  - Anwendungsschicht (HTTP, SSH, SMTP, IMAP, ...)
  - Transportschicht (TCP, UDP)
  - Internetschicht (IPv4, IPv6)
  - Netzzugangsschicht (Ethernet, WLAN)
- hier verwendet: sog. TCP/IP-Schichtenmodell

Anwendungsschicht

- Protokolle in dieser Schicht legen fest, wie Daten auf Anwendungsebene verpackt werden
- Beispiel: Kommunikation zwischen Browser und Webserver

Transportschicht

- nimmt Daten von der Anwendung entgegen und teilt diese passend für die Übertragung auf
- TCP (Transmission Control Protocol) sorgt dafür, dass alle Daten in der richtigen Reihenfolge ankommen (ggf. wiederholte Übertragung)
- TCP geeignet für z. B. HTTP, SSH (Datenverlust unerwünscht)
- UDP (User Datagram Protocol) garantiert keine Vollständigkeit oder die richtige Reihenfolge
- UDP geeignet für z. B. Voice over IP (kleinere Fehler fallen nicht auf, Verzögerungen bei Neuübertragung aber sehr)

Ports

- TCP-Segmente und UDP-Datagramme enthalten Quell- und Ziel-Port
- Kernel kann Port-Nummer einer Anwendung zuordnen und übergibt dieser die enthaltenen Daten
- einige Ports:
  - 22/TCP: SSH (Secure Shell)
  - 25/TCP: SMTP (Simple Mail Transfer Protocol)
  - 53/UDP, 53/TCP: DNS (Domain Name System)
  - 80/TCP: HTTP (Hypertext Transfer Protocol)
- insgesamt gibt es 2^{16} (65536) TCP- und UDP-Ports
- Server melden ihr Interesse an ankommenden Daten an, indem sie auf einem Port „lauschen“ (listen)

netstat

- `netstat` kann Netzwerkverbindungen und Serverdienste auflisten
- Wichtige Optionen:
  - `-t`: TCP-Verbindungen und -Server anzeigen
  - `-u`: UDP-Server anzeigen
  - `-l`: Server statt Verbindungen anzeigen
  - `-p`: Names und PID des Programms mit ausgeben
  - `-e`: mehr Informationen ausgeben
  - `-n`: nicht versuchen, den Hostname zu IP-Adressen zu ermitteln (Ausgabe erfolgt schneller)
- leicht zu merken: `netstat -tulpen`

Internetschicht

- überträgt Daten vom Sender zum Empfänger über das Internet
- Daten werden mit Quell- und Ziel-IP-Adresse versehen

Netzzugangsschicht

- Übertragung von Daten zwischen direkt (oder über Switch/Hub) verbundenen Rechnern
- legt fest, wie Daten zur Übertragung über physikalische Medien kodiert werden
- Protokolle: IEEE 802.3 (Ethernet), IEEE 802.11 (WLAN)
- nutzen physikalische Adresse der Netzwerkkarte (MAC-Adresse) zur Adressierung

IP-Adressen

- um Daten an einen anderen Rechner zu senden, braucht dieser eine global eindeutige Adresse
- IP (Internet Protocol) legt Adressformat fest

IPv4

- Adressen sind 32-Bit-Zahlen, die in viermal acht Bit aufgeteilt werden und durch „.“ getrennt dezimal dargestellt werden
- Beispiel: `198.51.100.27`
- theoretisch bis zu 2^{32} (ca. 4,3 Mrd.) Adressen
- viele Adressen reserviert (private Adressen, Dokumentation, Spezialanwendungen)
- anfangs wurden Adressen sehr großzügig vergeben
- Folge: kaum noch frei IPv4-Adressen

IPv6

- Nachfolger von IPv4
- Adressen sind 128-Bit-Zahlen, die in acht mal 16 Bit aufgeteilt werden und durch „:“ getrennt hexadezimal dargestellt werden
- Beispiel: `2001:db8:1234:dead:beef:cafe:5678:29`
- theoretisch bis zu 2^{128} (ca. $3,4 \cdot 10^{38}$) Adressen
- Anmerkung: mehrere aufeinanderfolgene Nullen werden bei IPv6-Adressen durch „::“ ersetzt, also
  `2001:db8:1234::27` statt
  `2001:db8:1234:0000:0000:0000:0000:27` (nur einmal pro Adresse)

Netzwerkinterfaces

- repräsentieren Netzwerkkarten
- damit ein Rechner unter einer IP-Adresse erreichbar ist, muss diese einem Netzwerk-Interface hinzugfügt werden
- Ethernet-Interface heißen `ethX` (fortlaufend ab 0 durchnummeriert), WLAN-Interfaces `wlanX` (kann bei manchen Herstellern abweichen, Atheros: `athX`)
- Spezial-Interface `lo` (Loopback) erlaubt es, über die IP-Adresse `127.0.0.1` bzw. `::1` auf den lokalen Rechner zuzugreifen
- Netzwerk-Interfaces haben keine Dateien in `/dev`

ifconfig

- zeigt die Konfiguration von Netzwerkinterfaces (IP-Adresse, Subnetzmaske) an und ändert sie
- `ifconfig` zeigt alle aktiven Netzwerkinterfaces an (mit `-a` auch nicht aktive)
- `ifconfig eth0` zeigt die Konfiguration von `eth0` an

Routing

- im Internet sind nicht alle Rechner direkt miteinander verbunden
- Internet besteht aus vielen Teil-Netzen
- Rechner in einem Netz können direkt miteinander kommunizieren
- zwischen zwei Netzen können Pakete über Router ausgetauscht werden

Subnetzmasken (IPv4) I

- legt fest, welche Rechner direkt erreichbar sind
- zeigt an, welche Bits in der IP-Adresse zum (festen) Netzwerk-Teil gehören und welche zum (variablen) Host-Teil
- Subnetzmaske hat das gleiche Format wie eine IP-Adresse
- alle festen Bits sind auf 1 gesetzt, alle variablen auf 0
- erste Adresse (alle Host-Bits auf 0) ist die Netzwerk-Adresse
- letzte Adresse (alle Host-Bits auf 1) ist die Broadcast-Adresse (Spezial-Adresse, um eine Nachricht an alle Rechner im Netz zu senden)

Subnetzmasken (IPv4) II

- Beispiel 1: IP-Addresse `192.0.2.50`, Subnetzmaske `255.255.255.0`
  - letzten 8 Bit der Adresse sind Host-Teil
  - direkt erreichbar sind die Adressen `192.0.2.1` bis `192.0.2.254`
  - `192.0.2.0` ist die Netzwerkadresse
  - `192.0.2.255` ist die Broadcast-Adresse
- Beispiel 2: IP-Addresse `192.0.2.50`, Subnetzmaske `255.255.255.128`
  - letzten 7 Bit der Adresse sind Host-Teil
  - direkt erreichbar sind die Adressen `192.0.2.1` bis `192.0.2.126`
  - `192.0.2.0` ist die Netzwerkadresse
  - `192.0.2.127` ist die Broadcast-Adresse

Präfixlängennotation (IPv6)

- anstatt eine Netzmaske mit allen gesetzten Bits anzugeben, wird bei IPv6 die Anzahl der gesetzten Bits durch „/“ getrennt an die Adrese angehangen
- Beispiel: `2001:db8:1234:dead:beef:cafe:5678:29/64`
- 64 Bits Host-Teil, 64-Bits Netzwerkteil
- zum Netz gehören die Adressen `2001:db8:1234:dead::` bis `2001:db8:1234:dead:ffff:ffff:ffff:ffff`
- die Präfixlängennotation kann auch bei IPv4 verwendet werden, viele Programme unterstützen diese aber nicht

Routing-Tabelle

- enthält Einträge für alle erreichbaren Netze und die Router über die diese erreicht werden (oder ob sie direkt erreichbar sind)
- auf normalen Rechner meistens nur ein Eintrag für das lokale Netz und ein Default-Gateway für alle anderen
- auf Routern oft sehr umfangreich
- empfängt ein Router ein Paket, ermittelt er den nächsten Router anhand seiner eigenen Routing-Tabelle
- keinem Rechner/Router im Internet muss daher der ganze Pfad zu allen anderen Rechnern bekannt sein
- Weg eines Pakets ergibt sich damit dynamisch während der Übertragung

route

- `route` zeigt die IPv4-Routing-Tabelle an
- zu jedem Ziel-Netz wird u. a. der nächste Router (Gateway) und das ausgehende Interface angezeigt
- `0.0.0.0` als Ziel-Netz ist der Eintrag für das Default-Gateway
- bei direkt erreichbaren Netzen ist `0.0.0.0` als Gateway angegeben
- `route -6` zeigt die IPv6-Routing-Tabelle an
- `::` bzw. `::/0` entspricht `0.0.0.0`

Domain Name System (DNS)

- IP-Adressen sind für Menschen schwer zu merken
- Alternative: Domain-Namen
- Beispiel: `www.unix-ag.uni-kl.de` statt `131.246.124.83` oder `2001:638:208:ef34:0:ff:fe00:83`
- DNS wird verwendet um Domain-Namen zu IP-Adressen aufzulösen
- dazu Anfrage an DNS-Server

/etc/resolv.conf

- Konfigurationsdatei, die die verwendeten DNS-Server und den Suchpfad enthält
- wird versucht, einen Rechnername ohne vollständige Domain aufzulösen, wird automatisch die im Suchpfad angegebenen Domain angehangen
- Suchpfad kann auch mehrere Domains enthalten (werden dann nacheinander ausprobiert)

Beispiel für /etc/resolv.conf

```
nameserver 198.51.100.15
nameserver 203.0.113.3
search example.com
```

- zur Namensauflösung wird zuerst `198.51.100.15` angefragt, wenn dieser nicht antwortet `203.0.113.3`
- der Suchpfad besteht aus `example.com`
- beispielsweise kann statt `www.example.com` auch einfach `www` angegeben werden

dig

- `dig` kann zum manuellen Auflösen von Rechnernamen verwendet werden
- `dig A www.example.com` fragt die IPv4-Adresse von `www.example.com` ab
- `dig AAAA www.example.com` fragt die IPv6-Adresse von `www.example.com` ab

Dynamic Host Configuration Protocol (DHCP)

- mit `ifconfig`, `route` und `/etc/resolv.conf` kann das Netzwerk von Hand konfiguriert werden
- schon bei wenigen Rechnern aufwendig und fehleranfällig
- Alternative: DHCP
- DHCP-Client fragt einen DHCP-Server nach den Netzwerk-Konfigurationsdaten (IP-Adresse, Netzmaske, Default-Gateway, DNS-Server, Suchpfad) und konfiguriert das Netzwerk entsprechend
- Standard bei den meisten Linux-Distributionen

Stateless address autoconfiguration (SLAAC)

- bei IPv6 wird DHCP üblicherweise nicht verwendet
- stattdessen beziehen die Rechner die Netzwerk-Konfiguration von Routern
- IP-Adresse wird vom Rechner aus der Netzwerk-Adresse und der MAC-Adresse gebildet
- DNS-Server über SLAAC zu konfigurieren wird noch nicht vollständig unterstützt

ping

- `ping` testet ob ein Rechner erreichbar ist
- sendet dazu jede Sekunde ein spezielles Paket an den Rechner
- kommt eine Antwort, wird diese angezeigt
- kommt keine Antwort, ist der Rechner nicht erreichbar
- als Argument kann eine IP-Adresse oder ein Rechnername angegeben werden
- `ping6` für IPv6-Adressen

Troubleshooting I

- wichtig zur Eingrenzung von Netzwerk-Problemen: systematisch vorgehen
- Schritt 1: Konfigurationsdaten sammeln:
  - IP-Adresse(n)
  - Netzmaske
  - Default-Gateway
  - DNS-Server und Suchpfad

Troubleshooting II

- Schritt 2: Können Rechner im lokalen Netz erreicht werden?
- Default-Gateway pingen
- antwortet das Default-Gateway nicht, sollte überprüft werden, ob
  - das Netzwerkkabel richtig steckt
  - IP-Adresse, Netzmaske und Gateway richtig konfiguriert sind

Troubleshooting III

- Schritt 3: Können Rechner außerhalb des lokalen Netzes erreicht werden?
- einen Rechner außerhalb des Netzes pingen, idealerweise eine IP-Adresse angeben, sodass DNS-Probleme ausgeschlossen werden können
- Beispiel: `ping 8.8.8.8` (öffentlicher DNS-Server von Google)
- kommt keine Antwort, sollte das Default-Gateway überprüft werden

Troubleshooting IV

- Schritt 4: Funktioniert der DNS-Server?
- DNS-Server pingen
- mit `dig` versuchen, einen Rechnernamen aufzulösen

Troubleshooting V

- Schritt 5: Laut um Hilfe rufen ;-)

ssh

- `ssh` ermöglicht es, über das Netzwerk eine Shell auf einem anderen Rechner zu starten
- Argument: `benutzername@anderer.rechner`
- alle Daten werden dabei verschlüsselt übertragen

Spickzettel

Alle Befehle

Befehl	Optionen
`netstat`	`-t -u`, `-l`, `-p`, `-e`, `-n`
`ifconfig`	`-a`
`route`	`-6`
`dig`	(`A`, `AAAA`)
`ping`	
`ssh`	