Netzwerk

Linux-Kurs der Unix-AG

Andreas Teuchert

7. Februar 2014

Netzwerk-Protokolle

- legen fest, wie Daten zur Übertragung verpackt werden
- unterteilt in verschiedene Schichten:
  - Anwendungsschicht (z. B. HTTP, SSH, SMTP)
  - Transportschicht (z. B. TCP, UDP)
  - Internetschicht (z. B. IPv4, IPv6)
  - Netzzugangsschicht (z. B. Ethernet, WLAN)
- OSI-Schichtenmodell definiert insgesamt sieben Schichten

Veranschaulichung: Versenden eines Briefes

1. Brief schreiben (in Deutsch, Englisch, ...)
2. lange Briefe auf mehrere Seiten/Blätter aufteilen
3. Brief in einen Umschlag stecken, Empfänger- und Absender-Adresse draufschreiben und Briefmarke draufkleben
4. Brief in einen Postbriefkasten werfen
5. Postbriefkasten wird irgendwann geleert und Brief zum Empfänger transportiert

- Daten durch das Internet versenden entspricht eher einer Postkarte: Jeder auf dem Weg kann alles lesen und verändern
- Abhilfe: Kryptografisch gesicherte Protokolle (HTTP**S**, SSH)

Beispiel: Aufrufen einer Webseite

1. Webbrowser erstellt eine HTTP-Anfrage
   - HTTP (Hypertext Transfer Protocol): „Sprache“ mit der sich Webbrowser und -Server unterhalten
2. HTTP-Anfrage wird in TCP-Segmente unterteilt
   - TCP (Transmission Control Protocol) sorgt dafür, dass alles in der richtigen Reihenfolge ankommt („Einschreiben“)
3. TCP-Segmente werden in IP-Pakete verpackt
   - IP-Pakete (Internet Protocol) enthalten global eindeutige Absender- und Empfänger-Adressen und können dadurch über das Internet verschickt werden
4. IP-Pakete werden in Ethernet-Frames verpackt und diese an das Default-Gateway gesendet
5. Das IP-Paket wird solange von Router zu Router weitergeleitet, bis es beim Webserver ankommt
6. Webserver packt die HTTP-Anfrage aus

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 freie 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:db8:1234:5678:21c:25ff:fe7c:d22b`
- 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: `athX`, `ethX`)
- manche Distributionen (z. B. Fedora) verwenden abweichende Benennung
- 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

Serverdienste

- Serverdienste (z. B. Webserver) stellen Dienste über das Netzwerk bereit
- warten dazu auf Verbindungen auf einem bestimmten Port (engl. „listen“)
- Port vergleichbar mit einer Wohneinheit in einem Mehrfamilienhaus
- nur ein Dienst pro Port
- insg. jeweils 65536 TCP- und UDP-Ports
- manche Ports sind fest vergeben (z. B. 80/TCP: HTTP-Webserver, 22/TCP: SSH)
- Client (z. B. Webbrowser) verbindet sich zum entsprechenden Port der Server-IP-Adresse

netstat

- `netstat` kann Netzwerkverbindungen und Serverdienste auflisten
- Wichtige Optionen:
  - `-t`: **T**CP-Verbindungen und -Server anzeigen
  - `-u`: **U**DP-Server anzeigen
  - `-l`: Server statt Verbindungen anzeigen (**l**istening)
  - `-p`: Names und PID des **P**rogramms mit ausgeben
  - `-e`: mehr Informationen ausgeben (**e**xtend)
  - `-n`: nicht versuchen, den Hostname zu IP-Adressen zu ermitteln (Ausgabe erfolgt schneller) (**n**umeric)
- leicht zu merken: `netstat -tulpen`
- alternativ: `lsof -i`

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
- welche Netze direkt erreicht werden können und welche Netze über welchen Router erreicht werden können, steht in der sog. Routing-Tabelle
- auf normalen Rechner meistens nur ein Eintrag für das lokale Netz und ein Default-Gateway für alle anderen

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
- Lösung: 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

/etc/hosts

- in `/etc/hosts` können eigene Domain-Einträge eingetragen werden
- z. B. für nur im lokalen Netz verwendete Hostnames oder um IP-Adressen von bestehenden DNS-Einträgen zu überschreiben
- Einträge gelten nur auf dem lokalen Rechner
- das Beispiel unten legt fest, dass der Rechner namens „heimserver.example“ oder kurz „heimserver“ unter der IP-Adresse `192.168.4.2` zu erreichen ist

```
192.168.4.2 heimserver.example heimserver
```

Automatische Netzwerk-Konfiguration

- mit `ifconfig`, `route` und `/etc/resolv.conf` kann das Netzwerk von Hand konfiguriert werden
- schon bei wenigen Rechnern aufwendig und fehleranfällig
- Alternative: Dynamic Host Configuration Protocol (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
- IPv6-Konfiguration üblicherweise per stateless address autoconfiguration (SLAAC) statt DHCP
- SLAAC bei allen modernen Betriebssystemen automatisch aktiv

ping

- `ping` sendet Test-Pakete an den angegebenen Rechner und zeigt an, ob eine Antwort ankam und wie schnell
- es werden solange Pakete gesendet, bis `ping` mit `Ctrl-C` beendet wird
- erwartet einen Rechnernamen oder eine IP-Adresse als Argument
- `ping6` verwendet IPv6 statt IPv4
- nützlich um Netzwerkprobleme zu untersuchen

Spickzettel

Alle Befehle

Befehl	Optionen
`ifconfig`	`-a`
`netstat`	`-t -u`, `-l`, `-p`, `-e`, `-n`
`route`	`-6`
`dig`	(`A`, `AAAA`)
`ping`, `ping6`	