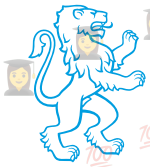


Kantonsschule Im Lee

Informatik

# Skript

Cyril Wendl

# Inhaltsverzeichnis

# Kapitel 1

# Netzwerke

## 1.1 Netzwerke und das Internet

Bevor wir uns mit dem Internet als spezielles Netzwerk befassen, wollen wir allgemein nochmals auf Netzwerke eingehen. Im Alltag treffen Sie Netzwerke an unterschiedlichsten Orten an, zum Beispiel:

- Transport-Netzwerke: Die Post, DHL, Planzer, etc.
- Soziale Netzwerke: TikTok, Youtube, Instagram, etc.
- Mobilitäts-Netzwerke: Das Strassennetz, SBB, Flixbus, etc.
- Shopping-Netzwerke: Aliexpress, Uber, etc.
- Computer-Netzwerke: **Das Internet**, Firmen-Netzwerke etc.

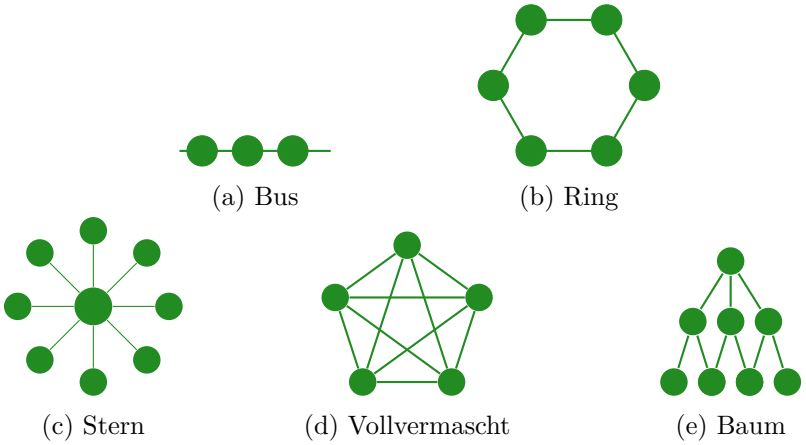
All diese Netzwerke können als Graphen visualisiert und veranschaulicht werden. Gemeinsam ist allen Netzwerken, dass die einzelnen Knoten ihren „Wert“ (ihren Nutzen) erst durch die Verbindung mit anderen Kanten erhalten. Anders gesagt, mit den Worten des „Erfinders des Internets“, Tim Berners-Lee:

> The web is more a social creation than a technical one.

Häufig werden die Begriffe “Internet” und “Computer-Netze”, bzw. “Rechner-Netze” austauschbar verwendet. Dabei ist es jedoch wichtig zu verstehen, dass dies zwei unterschiedliche Konzepte sind: während dem mit dem *Internet* ein höchst komplexes, den gesamten Globus umspannendes Netzwerk von Computern gemeint ist, können lediglich zwei miteinander verbundene Computer bereits ein Rechnernetz bilden. Das Internet wird häufig auch als *Netz von (Rechner-)Netzen* bezeichnet. Es ist daher wichtig, dass wir uns zuerst mit einfacheren Rechnernetzen befassen, da diese die Grundbausteine für das Internet bilden.

### 1.1.1 Topologie

Beim Erstellen eines Netzwerks stellt sich als Erstes die Frage, welche Computer man mit welchen anderen Computern verbinden soll. Die Struktur der Verbindungen in einem Netzwerk kann also umformuliert werden als Graphen-Problem, bei dem es zu bestimmen gibt, welche Knoten eines Graphen miteinander über eine Kante verbunden werden. Die verschiedenen möglichen Strukturen eines Netzwerks werden auch als **Topologie** bezeichnet. Einige Beispiel-Topologien sind in **??** gezeigt.



(a) Bus (b) Ring

(c) Stern (d) Vollvermascht (e) Baum

Abbildung 1.1: Beispiele von Netzwerk-Topologien
ig:topologie

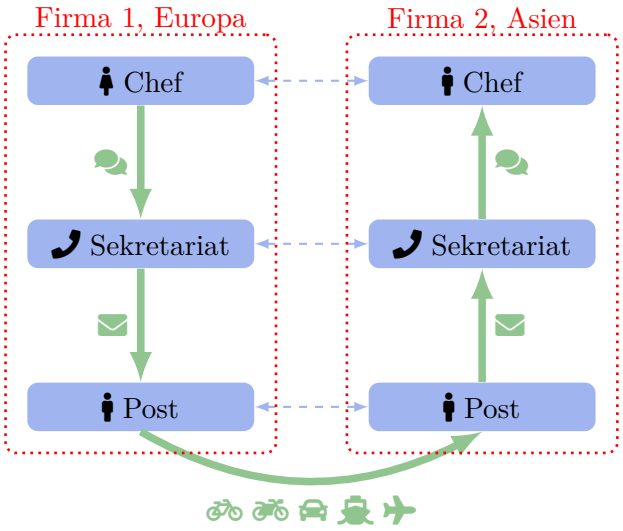
Da Computer in Realität häufig nur über eine einzige Netzwerkkarte verfügen, können sie auch nur mit einem weiteren Computer verbunden werden. In einem lokalen Netzwerk trifft man daher häufig die Form einer Stern-Topologie an, bei der alle Computer mit einer zentralen **Switch** verbunden werden, also einem Gerät, welches mehrere Rechner an einem zentralen Punkt miteinander verbindet.

## 1.2 Internet-Schichten

### 1.2.1 Schichtenmodell: Analogie

Die Kommunikation über das Internet verläuft in unterschiedlichen Teilschritten. Man spricht von dem *Schichtenmodell*. Als Analogie zum Schichtenmodell sei folgende Situation gegeben, in der die Chefs zweier unterschiedlicher Firmen miteinander kommunizieren wollen (siehe **??**):

1. Chef 1 (, in Europa) möchte mit Chef 2 (, in Asien) sprechen.
2. Dazu übermittelt Chef 1 dem Sekretariat Ihre Nachricht an Chef 2. Das Sekretariat schreibt die Nachricht nieder und übermittelt der lokalen Post den Brief an Chef 2.
3. Die lokale Post kümmert sich nun um den Transport zur asiatischen Post, wo der Brief an das Sekretariat von Firma 2 übergeben und schlussendlich an Chef 2 gelangt.




ig:schichtenmodell$_a$*nalogie*

Abbildung 1.2: Analogie zum Schichtenmodell: Kommunikation zwischen zwei grossen Firmen

Dieser Kommunikationsablauf kann in Schichten mit jeweiligen dazugehörigen Prozessen aufgeteilt werden:

1. **Schicht "Chefs"**: damit die beiden Chefs mit einander sinnvolle Gespräche führen können, verfügen Sie über gemeinsame Standards (Sprache, Wissen etc.) sowie Protokolle (standardisierte Abläufe der Begrüssung etc.).
2. **Schicht "Sekretariat"**: Damit die Sekretariate miteinander kommunizieren können, benötigen auch sie standardisierte Abläufe oder Protokolle, etwa, wie man einen Brief korrekt verfässt, datiert und addressiert.
3. **Schicht "Post"**: Die Post verfügt ebenfalls über standardisierte Protokolle (wie werden Briefe und Pakete von A nach B transportiert? Mit welchen Transportmitteln, über welche Routen?)

Die Vorteile des Schichtenmodells scheinen offensichtlich:

- **Unabhängigkeit der Schichten**: Falls die Post das Transportmittel ändert, müssen die Sekretariate die Briefe nicht neu verfassen.
- **Modularität und Spezialisierung**: Statt das jemand alles tun muss (Chef 1 reist nach Asien zu Chef 2), kann der Prozess für die jeweiligten Beteiligten schnell und effizient abgewickelt werden.

### 1.2.2 TCP-IP- und OSI-Schichtenmodell

Im Zusammenhang mit dem Internet werden häufig zwei Schichtenmodelle gegenübergestellt: Das vollständige, detaillierte Open Systems Interconnection (OSI)-Modell mit 7 Schichten sowie das etwas vereinfachte Transmission Control Protocol (TCP)-Internet Protocol (IP)-Modell mit 4 Schichten. Die beiden Schichten, deren Aufgaben, Informationsformen und Protokolle sind in **??** abgebildet.

| # | Schicht (OSI) | Schicht (TCP / IP) | Aufgabe | Informationsform | Gerät / Protokoll |
|---|---|---|---|---|---|
| 7 | Anwendung *Application* | | Hilft bei der Identifizierung des Clients und synchronisiert die Kommunikation. | Nachricht | **HTTP(S)**, FTP, SMTP... |
| 6 | Darstellung *Presentation* | Anwendung *Application* | Daten von der Anwendungsschicht werden extrahiert und für die Übertragung in das erforderliche Format gebracht. | Nachricht | JPEG, MPEG, GIF, **HTML**... |
| 5 | Sitzung *Session* | | Stellt Verbindung her, hält diese aufrecht, gewährleistet Authentifizierung und sichert die Sicherheit. | Nachricht (oder verschlüsselte Nachricht) | Gateway, NetBIOS, PPTP, ... |
| 4 | Transport *Transport* | Transport *Transport* | Nimmt Dienst von der Vermittlungsschicht und stellt ihn der Anwendungsschicht zur Verfügung. | Segment | Firewall, **TCP**, UDP, Ports... |
| 3 | Vermittlung *Network* | Internet *Internet* | Übertragung von Daten von einem Host zu einem anderen, der sich in unterschiedlichen Netzwerken befindet. | Paket | Router, **IPv4**, IPv6 |
| 2 | Sicherung *Data Link* | Netzzugriff *Data Link* | Übermittlung von Nachrichten von Knoten zu Knoten. | Frame | Switch, Bridge, **MAC**... |
| 1 | Bitübertragung *Physical* | | Herstellung physischer Verbindungen zwischen Geräten. | Bits | Hub, Repeater, Modem, Kabel... |

able:osi-layers

Tabelle 1.1: Internet-Schichten-Modelle Open Systems Interconnection (OSI) und Transmission Control Protocol (TCP)-Internet Protocol (IP) sowie deren wichtigste Aufgaben, Informationsformen, Geräte und Protokolle

Im Folgenden gehen verwenden wir hauptsächlich das etwas vereinfachte Transmission Control Protocol (TCP)-Internet Protocol (IP)-Modelle und gehen kurz auf jede Schicht ein.

**Aufgabe 1.1**

Um sich auf die Vertiefung in die jeweiligen Schichten und Protokolle einzustimmen, schauen Sie sich folgendes Video an:

Youtube-Video (13')

Notieren Sie sich dabei Ihre Antworten auf folgende Fragen:

- Welche **Schichten**, **Geräte** und **Protokolle** werden genannt?
  - Namen (Abkürzung und ganz)
  - Wofür sind sie zuständig?
  - Welche Protokolle gehören zu welchen Schichten?
  - Welche Geräte gehören zu welchen Protokollen / Schichten?
- Notieren Sie Ihre **Fragen**: Welche Begriffe verstehen Sie (noch) nicht?

**Aufgabe 1.2**

Tauschen Sie sich nun in 3er- bis 4er-Gruppen aus und tragen Sie ihre Notizen zusammen. Beantworten Sie danach folgende Fragen:

- Was ist eine URL?
- Was macht "Mr. IP"?
- Was macht der Router?
- Was macht die Firewall?
- Was macht der Proxy Server?
- Was macht der "Ping of Death"? 💀

### 1.2.3 Anwendungsschicht

Bevor wir uns näher mit den Dimensionen der Anwendungsschicht sowie der darunterliegenden Schichten befassen, sollten wir zuerst einige häufig auftauchende Begriffe klären, von denen im Kontext von Rechner-Netzen häufig die Rede ist. Je nach Rolle der Computer in einem Rechner-Netz ist häufig von **Client**, **Server** oder **Host** die Rede:

- Der **Client** $\mathcal{C}$ (von en. *client* = "Kunde") bezeichnet den Computer $\mathcal{C}$, der etwas von einem anderen Computer $\mathcal{S}$ will. Genauer gesagt handelt es sich nicht um den Computer selber, sondern um ein Programm auf einem Computer $\mathcal{C}$, das etwas anfordert von einem anderen Programm, das auf einem anderen Computer $\mathcal{S}$ läuft. Zur Vereinfachung sprechen wir jedoch häufig einfach von einem Computer $\mathcal{C}$ und einem Computer $\mathcal{S}$.
- Der **Server** $\mathcal{S}$ (von en. *to serve* = "dienen") bezeichnet den Computer $\mathcal{S}$ (genauer gesagt das Computer-Programm $\mathcal{S}$), das dem Computer $\mathcal{C}$ gibt, was er will. Der Server kann Verbindungsanfragen von anderen Computern akzeptieren oder ablehnen.
- Bei einem **Host** (von englisch *host* = "Wirt" / "Gastgeber") ist im allgemeinen ein Computer in einem Netzwerk gemeint, der je nach Situation die Rolle des Servers oder Clients einnehmen kann. Der Name eines Computers in einem Netzwerk wird häufig als **Host Name** bezeichnet.

### 1.2.4 Transportschicht

In der Transportschicht geht es, wie es der Name bereits sagt, darum, wie Daten von einem Host zum nächsten transportiert werden.

Die Daten, die zwischen zwei Computern übertragen werden, werden häufig auch als **Nutzdaten** oder *payload* bezeichnet. Zu diesen Nutzdaten, die zwischen zwei Computern hin- und hergesandt werden sollen, kommen nun noch verschiedene weitere Daten dazu, so genannte **Metadaten**. Metadaten sind dazu zuständig, wichtige Informationen zu den Ursprungsdaten hinzuzufügen, in diesem Falle Daten, um den Transport der Daten zu ermöglichen. Diese Metadaten werden vom Übertragungs-Steuerungs-Protocoll (Transmission Control Protocol (TCP)) zur Verfügung gestellt und genutzt, in einem so genannten *Header* (s. **??**).

| **TCP Header** | |
|---|---|
| Source Port | Destination Port |
| Sequence # (SEQ) | |
| Acknoledgement # (ACK) | |
| Weitere TCP-Header-Felder | |
| **Nutzdaten** | |
| Anwendungsdaten | |

Abbildung 1.3: TCP-Header (vereinfachte Darstellung)
ig:tcp-header

Einige der wichtigsten Metadaten sind:

- **Der Port** (von engl. *port* = Hafen): Dies ist eine Zahl, mit der der Computer weiss, welche Anwendung (z.B. Mail, Browser, Gaming-Programm etc.) Daten verschickt. Ein Computer kann, wie Sie vermutlich wissen, gleichzeitig über mehrere Programme Daten von anderen Computern empfangen und an sie schicken: sie können beispielsweise gleichzeitig eine Nachricht über WhatsApp schicken und ein Video schauen. Damit der Computer weiss, welche Anwendung die Daten verschickt, verseht er die Daten typischerweise mit einem Port. Obschon Ports für jedes Programm frei gewählt werden können, werden für bekannte Protokolle typischerweise Standard-Portnummern verwendet (s. **??**).

| **Portnummer** | **Anwendung** | **Verwendung** |
|---|---|---|
| 20 | File Transfer Protocol (FTP) | Dateitransfer |
| 21 | File Transfer Protocol (FTP) | Befehlssteuerung |
| 22 | Secure Shell (SSH) | Sichere Shell-Zugriffe |
| 23 | Telnet | Unverschlüsselte Fernsteuerung |
| 25 | Simple Mail Transfer Protocol (SMTP) | E-Mail-Versand |
| 53 | Domain Name System (DNS) | Namensauflösung |
| 80 | Hypertext Transfer Protocol (HTTP) | Webseiten-Abruf |
| 110 | Post Office Protocol Version 3 (POP3) | E-Mail-Abholung |
| 143 | Internet Message Access Protocol (IMAP) | E-Mail-Management |
| 443 | Hypertext Transfer Protocol Secure (HTTPS) | Verschlüsselter Webseiten-Abruf |
| 993 | Internet Message Access Protocol (IMAP) | E-Mail-Management über SSL |
| 995 | Post Office Protocol Version 3 (POP3) | E-Mail-Abholung über SSL |

Tabelle 1.2: Gängige TCP-Ports, Anwendungen und deren Verwendung
ab:tcp$_{p}orts$

Sowohl der *Client* wie auch der *Server* versehen die Daten mit jeweils einer Zahl, die für den

Ursprungs-Port (*Source Port*), respektive den Ziel-Port (*Destination Port*) stehen.
- **Ziffern** für die *Sequence* (Sequenz) und *Acknowledgement* (Anerkennung) von Paketen, die die richtige Reihenfolge der gesendeten und erhaltenen Daten gerantieren.
- Weitere Daten, die für den Transport notwendig sind.

**Aufgabe (Challenge) 1.3**

Lösen Sie die Socket- und Port-Aufgaben auf folgendem Link: https://www.inf-schule.de/rechnernetze/anwendung/socketprogrammierung/Anfragen_an_einen_Server_Stellen.

Zusätzlich zur Steuerung der Datenübertragung können TCP-Metadaten genutzt werden, um die erste Verbindung zwischen zwei Computern herzustellen. Dies erfolgt typischerweise über das 3-Wege-Handschlag-Protokoll (s. **??**).

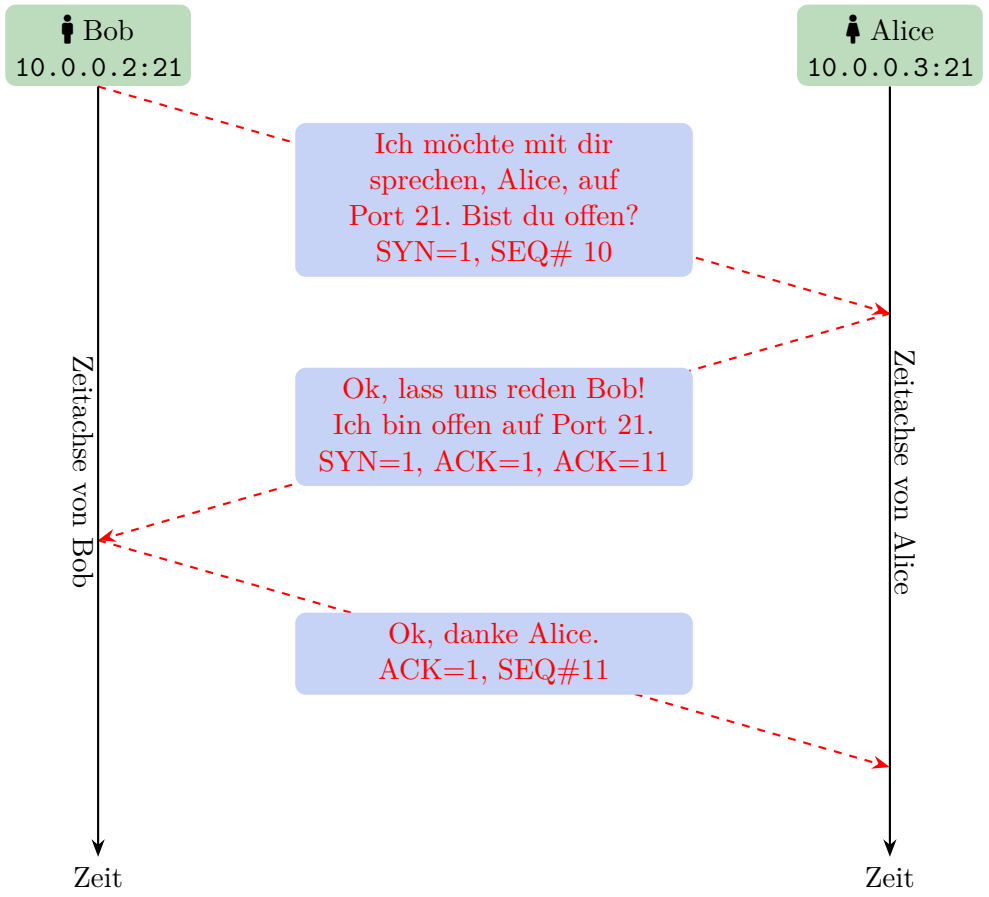



Abbildung 1.4: Drei-Wege-Handschlag im TCP-Protokoll
ig:threewayHandshake

Folgende Schritte des Drei-Wege-Handschlags werden in **??** illustriert:

1. Computer A (*client*, `10.0.0.2`) erstellt eine Verbindungsanfrage zum Computer B (*server*, `10.0.0.3`), mittels einem Paket, das nur das `SYN`-Flaggen gesetzt hat.
2. Der Server antwortet sowohl mit einem gesetzten `SYN` und einem `ACK`-Flaggen
3. Im letzten Schritt antwortet der *client* nochmals mit einem einzelnen `ACK`-Flaggen

Somit ist die TCP-Verbindung ist nun erstellt und die beiden Maschinen können miteinander kom-

munizieren.

### 1.2.5 Internetschicht

Die Übertragungssicherungsschicht befasst sich mit der stabilen Übertragung von Daten zwischen zwei Rechnern. Wofür sie jedoch nicht zuständig ist, ist die Adressierung und Identifizierung von Computern im Netzwerk. Diesem Aspekt ist die Internetschicht gewidmet.

#### 1.2.5.1 IP-Adressen

Eine Internet Protocol (IP)-Adresse ist eine Netzwerk-Adresse eines Computers. Somit ist sie in gewisser Weise der Wohnadresse einer Person ähnlich. Häufig werden IP-Adressen noch im **IPv4**-Format angegeben: Dieses besteht aus 4 bytes, also 4 Zahlen von 0 bis 255. Die IPv4-Adresse Wird meist dezimal angegeben (beispielsweise `192.168.0.1`).

**↑ Aufgabe (Abgabe) 1.4**

Finden Sie Ihre eigene IPv4-Adresse heraus. Tipps:

- MacOS, Terminal (via Spotlight-Suche):
  `ipconfig getifaddr en0`
- Windows, Programm "cmd":
  `ipconfig`

**Aufgabe 1.5**

Wie viele IPv4-Adressen gibt es?

**✓ Lösungsvorschlag zu Aufgabe 1.5**

- $4 \cdot 1$ byte $= 4 \cdot 8$ bits $= 32$ bits
- $2^{32} \approx 4$ Milliarden Adressen
- $\rightarrow$ zu wenig!

Wie wir in Aufgabe 1.5 gesehen haben, gibt es zu wenige Adressen bei IPv4. Daher wird IPv4 graduell durch einen neuen Standard abgelöst, **IPv6**, in welchem jede Adresse über 128 Bits verfügt, womit fast unendlich viele Geräte addressiert werden können.

**Aufgabe 1.6**

Wie viele Adressen gibt es bei IPv6?

**✓ Lösungsvorschlag zu Aufgabe 1.6**

$$2^{128} \approx 3.4 \cdot 10^{38}$$

#### 1.2.5.2 Ping

Mit dem Befehl "`ping`" kann überprüft werden, ob ein Rechner im Netzwerk erreichbar ist. Dabei wird eine minimale, inhaltslose Nachricht an einen anderen Rechner (=IP-Adresse) geschickt. Das Ziel ist es, zu überprüfen, ob eine gewisse IP-Adresse im Netzwerk existiert. Der Befehl "*ping*" ist

vergleichbar mit einem Ping-Pong-Spiel: Der Absender-Rechner, der eine "ping"-Nachricht schickt, erhält vom Ziel-Rechner, sofern dieser erreicht wird, viermal eine "pong"-Nachricht zurückgeschickt. Der `ping`-Befehl kann im Terminal (MacOS, UNIX-Systeme), bzw cmd (Windows) wie folgt ausgeführt werden: `ping [ip address]`, also z.B. `ping 159.233.1.22`.

**Achtung**

Eine Spezial-Variante des `ping`-Befehls ist der sogenannte Ping of Death, bei dem ein bösartiger Computer ein zu grosses Datenpaket an einen anderen Computer schickte, das beim Zusammensetzen zum Zusammensturz des Zielrechners führte. Dies ist seit Ende der 90er-Jahre auf den meisten Computern nicht mehr möglich: Die Sicherheitslücke wurde behoben, indem beispielsweise die Paketgrösse vor dem Zusammensetzen überprüft wurde.

**Aufgabe 1.7**

Installieren Sie das Programm Filius, welches wir zur Modellierung von Netzwerken verwenden werden. Die Anleitung dazu finden Sie in Unterabschnitt G.

**Aufgabe (Abgabe) 1.8**

Schauen Sie sich jeweils zuerst die Videos an und erledigen Sie danach Aufgaben 2.1 - 2.3 im Filius-Workshop: Link (Credits: Gymnasium Kirchenfeld)

→ **Abgabe `.fls`-Dateien auf Moodle!**

**Achtung**

Je nach Gerät ist die linke Seitenleiste nicht sichtbar und Sie können Aufgabe 2.2 daher nicht sehen. Klicken Sie auf "Themenübersicht" → "IN GYM2" → "Netzwerke" um die linke Seitenleiste korrekt zu sehen.

**Aufgabe (Challenge) 1.9**

Lösen Sie Aufgaben 1-6 unter diesem Link (Kapitel "Vernetzung von Rechnern").

#### 1.2.5.3 Netzmasken & Subnetze

**Subnetzwerke** verbinden — wie es der Name sagt— mehrere Hosts zu einem (Sub-)Netzwerk. Dabei gehören mehrere Geräte zum gleichen Subnetz, falls Sie:

- Physisch miteinander verbunden sind (WLAN, Kabel, etc.)
- Die gleiche Subnetzmaske teilen (später mehr dazu)

Eine (Sub-)Netzmaske hat fast dasselbe Format wie eine IP-Adresse und fasst mehrere IPs zu einer Gruppe, d.h. einem Subnetz zusammen. Die Subnetzmaske gibt an, wie viele Stellen einer IP-Adresse innerhalb eines Subnetzes gleich sein müssen. Sie besteht immer aus zwei IP-Adressen, wobei die erste IP-Adresse einer gültigen Adresse innerhalb des Subnetzes entspricht, und die zweite IP-Adresse vorgibt, wie stark eine weitere IP-Adresse von der ersten IP-Adresse abweichen dürfte, damit sie immer noch zum selben Netz gehört.

**Beispiel 1.1:**
Gegeben sei folgende Subnetzmaske:

| | Dezimal | Binär |
|---|---|---|
| IP-Adresse von Computer 1 | `159.233.1.22` | `10011111.11101001.00000001.00010110` |
| IP-Adresse von Computer 2 | `159.233.1.1` | `10011111.11101001.00000001.00000001` |
| Subnetzmaske | `255.255.255.0` | `11111111.11111111.11111111.00000000` |

Tabelle 1.3: Beispiel einer Subnetzmaske mit zwei IP-Adressen
able:ex-subnet

Um herauszufinden, ob die IP-Adressen von Computer 1 und Computer 2 zum selben Netzwerk gehören, muss überprüft werden, ob die binäre IP-Adresse an all denjenigen Stellen gleich ist, wo die Subnetzmaske = "1" ist (also an allen grün markierten Stellen in Tabelle **??**). In diesem Beispiel bedeutet dies, dass alle Internet Protocol (IP)-Adressen, welche mit `159.233.1.[...]` beginnen zum selben Netzwerk dazugehören.

Aufgabe 1.10

| **Eigene IP** | **Netzmaske** | **Ziel-IP** | **Gleiches Subnetz?** |
|---|---|---|---|
| 213.45.19.89 | 255.255.255.0 | 213.45.17.89 | |
| 213.45.19.89 | 255.255.0.0 | 213.45.17.89 | |
| 88.100.11.17 | 255.255.255.0 | 88.100.11.254 | |
| 88.100.11.17 | 0.0.0.0 | 213.45.19.89 | |
| 10.0.0.0 | 255.255.255.252 | 10.0.0.1 | |
| 1.2.3.0 | 255.255.255.252 | 1.2.3.5 | |

- → IPs dürfen sich nur an denjenigen Stellen unterscheiden, wo in der Maske (binär!) Nullen stehen.
- Umrechner dezimal → binär: https://oinf.ch/interactive/ips-und-netzmaske/

✔ Lösungsvorschlag zu Aufgabe 1.10

| **Eigene IP** | **Netzmaske** | **Ziel-IP** | **Gleiches Subnetz?** |
|---|---|---|---|
| 213.45.19.89 | 255.255.255.0 | 213.45.17.89 | Nein |
| 213.45.19.89 | 255.255.0.0 | 213.45.17.89 | Ja |
| 88.100.11.17 | 255.255.255.0 | 88.100.11.254 | Ja |
| 88.100.11.17 | 0.0.0.0 | 213.45.19.89 | Ja (alles ✔) |
| 10.0.0.0 | 255.255.255.252 | 10.0.0.1 | Ja ($252_{10} = 11111100_2$) |
| 1.2.3.0 | 255.255.255.252 | 1.2.3.5 | Nein ($252_{10} = 11111100_2$) |

## Aufgabe 1.11

Typischerweise beginnen alle IP-Adressen, die auf das lokale Netzwerk (=das Netzwerk, in dem sich der eigene Computer befindet) mit `192.168.xxx.xxx`. Die Subnetzmaske lautet also `255.255.0.0`. Wie viele Hosts haben in so einem Netzwerk Platz? Schreiben Sie die Subnetzmaske binär auf und rechnen Sie aus, wie viele unterschiedliche Geräte sich in diesem lokalen Netzwerk befinden können.

✔ Lösungsvorschlag zu Aufgabe 1.11

Zwei Bytes (die zwei letzten Byte-Gruppen) der Subnetzmaske sind reserviert für unterschiedliche Kombinationen von IP-Adressen. Es können sich daher bis zu $2^{16} = 65'536$ Hosts in einem solchen lokalen Netzwerk befinden.

## Aufgabe 1.12

Sechs Subnetze sind gegeben durch je eine Netzmaske und eine IP eines sich darin befindenden Rechners. Entscheiden Sie für jede links aufgeführte IP, zu welchen Subnetzen der entsprechende Rechner gehört.

| **Netzmaske Subnetz** | 255.255.255.0 | | 255.255.0.0 | | 255.255.255.240 | |
|---|---|---|---|---|---|---|
| IP | 3.4.5.6 | 3.3.3.3 | 3.4.5.6 | 3.3.3.3 | 3.4.5.6 | 3.3.3.3 |
| 3.3.3.4 | | | | | | |
| 4.4.4.3 | | | | | | |
| 3.4.5.99 | | | | | | |
| 3.3.4.4 | | | | | | |
| 3.4.7.7 | | | | | | |
| 3.3.3.17 | | | | | | |

Überprüfen: https://oinf.ch/interactive/ips-und-netzmaske/

✔ Lösungsvorschlag zu Aufgabe 1.12

| **Netzmaske Subnetz** | 255.255.255.0 | | 255.255.0.0 | | 255.255.255.240 | |
|---|---|---|---|---|---|---|
| IP | 3.4.5.6 | 3.3.3.3 | 3.4.5.6 | 3.3.3.3 | 3.4.5.6 | 3.3.3.3 |
| 3.3.3.4 | Nein | Ja | Nein | Ja | Nein | Ja |
| 4.4.4.3 | Nein | Nein | Nein | Nein | Nein | Nein |
| 3.4.5.99 | Ja | Nein | Ja | Nein | Nein | Nein |
| 3.3.4.4 | Nein | Nein | Nein | Ja | Nein | Nein |
| 3.4.7.7 | Nein | Ja | Ja | Nein | Nein | Nein |
| 3.3.3.17 | Nein | Ja | Nein | Ja | Nein | Nein |

#### 1.2.5.4 Router & Gateways

Bis anhin haben wir gesehen, wie wir einzelne Rechner mittels einer IP adressieren und wie wir diese mittels einer Subnetze zu einem Netzwerk verbinden. Was, wenn wir jedoch zwei Subnetze zu einem grossen Netzwerk verbinden wollen? Genau dies geschieht im Internet, welches im Wesentlichen aus einem Netzwerk von Netzwerken besteht. Um mehrere Netzwerke miteinander zu verbinden, verwenden wir einen **Router**. Diesen schauen wir uns in diesem Abschnitt genauer an.

Ähbnliche einer Haustür verbindet ein Router ein Subnetz (=Haus) mit der Aussenwelt, er befindet sich also an der Grenze zwischen zwei oder mehreren Subnetzen. Über die Routing-Tabelle kann ein Router entscheiden, auf welchen Weg ein Datenpaket an die Aussenwelt verschickt werden kann. Eine typische Routing-Tabelle enthält folgende Informationen:

- **Ziel-IP und Netzmaske**: Diese beiden Spalten definieren, welche Subnetze dem Router bekannt sind (sowohl eigenes Subnetz wie andere Subnetze).
- **Gateway** = Tor, Einfahrt: Der nächstgelegene Zielort, an den ein Paket geschickt werden

muss, um eine gewisse Ziel-IP zu erreichen. Das Gateway kann als eine Art "Tor zur Aussenwelt" angeschaut werden und ist häufig die IP-Adresse des Routers der Ziel-IP.
- **Interface** = Schnittstelle: Die physikalische Schnittstelle am Router (z.B. Ethernet-Stecker), über die eine Ziel-Adresse (Gateway) erreichbar ist.

Ein Beispiel einer Routing-Tabelle ist in **??** abgebildet.

| Ziel-IP | Netzmaske | Gateway | Interface |
|---|---|---|---|
| 0.0.0.0 | 0.0.0.0 | 192.168.1.1 | eth0 |
| 192.168.1.0 | 255.255.255.0 | 0.0.0.0 | eth0 |
| 192.168.2.0 | 255.255.255.0 | 0.0.0.0 | eth1 |
| 10.0.0.0 | 255.0.0.0 | 192.168.1.2 | eth0 |
| 172.16.0.0 | 255.240.0.0 | 192.168.2.2 | eth1 |
| 127.0.0.0 | 255.0.0.0 | 0.0.0.0 | lo |

Tabelle 1.4: Beispiel einer Routing-Tabelle
able:routing

**Aufgabe 1.13**

Lesen Sie mehr über die Aufgaben und Funktionsweise von Routern und Gateways, indem Sie diese Webseite genau durchlesen.

**Aufgabe (Abgabe) 1.14**

Schauen Sie sich jeweils zuerst die Videos an und erledigen Sie danach Aufgaben 2.4 - 2.6 im Filius-Workshop: Link. Laden Sie Ihre Lösung zu 2.6 als `.fls`-Datei auf Moodle hoch.

#### 1.2.5.5 Clients & Servers

Eine Ihnen bekannte Situation findet sich vor, wenn ein Client $\mathcal{C}$ eine Webseite von einem Server $\mathcal{S}$ anfordert. Dies geschieht beispielsweise, wenn Sie von Ihrem Mobiltelefon oder Laptop aus eine Webseite über den Browser aufrufen. Im Folgenden werden wir auf vereinfachte Weise veranschaulichen, welche Schritte geschehen müssen, damit Sie ihre Webseite sehen können.

1. Als erstes rufen Sie Ihren **Browser** (von en. *to browse* = "stöbern") auf, dies können beispielsweise sein: (a) Chrome (b) Firefox (c) Safari (d) Edge (e) etc. Dabei handelt es sich um Programme, die eine Datei (meist `.html`) aus dem Internet anfordern und diese nach dem Empfang darstellen können. Die angeforderten Dateien enthalten dabei Informationen zu Inhalten, Struktur, Darstellung und Interaktivität einer Webseite. Das Senden / Empfangen von Webseite-Dateien selber übernimmt das Betriebssystem (z.B. Windows, MacOS), nachdem es durch den Browser dazu aufgefordert wurde.
2. Sobald Sie Ihren Browser aufgerufen haben, geben Sie eine Web-Adresse ein. Eine solche Web-Adresse wird im Allgemeinen Uniform Resource Locator (URL) genannt und besteht aus folgenden Komponenten:

$$\underbrace{\text{http://}}_{\text{Protokoll}}\underbrace{\text{www.}}_{\text{Server}}\underbrace{\text{beispiel.}}_{\text{Domain}}\underbrace{\text{ch}}_{\text{TLD}}\underbrace{\text{/dokumente/}}_{\text{Ordner}}\underbrace{\text{reglemente.html}}_{\text{Dateiname}}$$

   - Der erste Teil (in diesem Fall `http://`) bestimmt das Protokoll, mit welchem die Datei vom Server angefordert wird.

- Der zweite Teil (`www.beispiel.ch`) bezeichnet dabei den *Server* $\mathcal{S}$, von dem wir die Datei anfordern. Dieser Name entspricht in der Regel einer einfachen IP-Adresse, wie beispielsweise `192.168.10.2`. Die Endung (`.ch`) steht für eine sogenannte Top-Level Domain (TLD), in diesem Fall wird durch "`.ch`" eine Schweizer Webseite bezeichnet.
- Alles nach der TLD ist ein gewöhnlicher Ordnerpfad auf dem Computer, der schlussendlich zur Datei "`reglemente.html`" führt, die den Inhalt der Webseite enthält

Der schematische Ablauf eines Webseiten-Abrufs mit dem HTTP-Protokoll ist in **??** abgebildet.

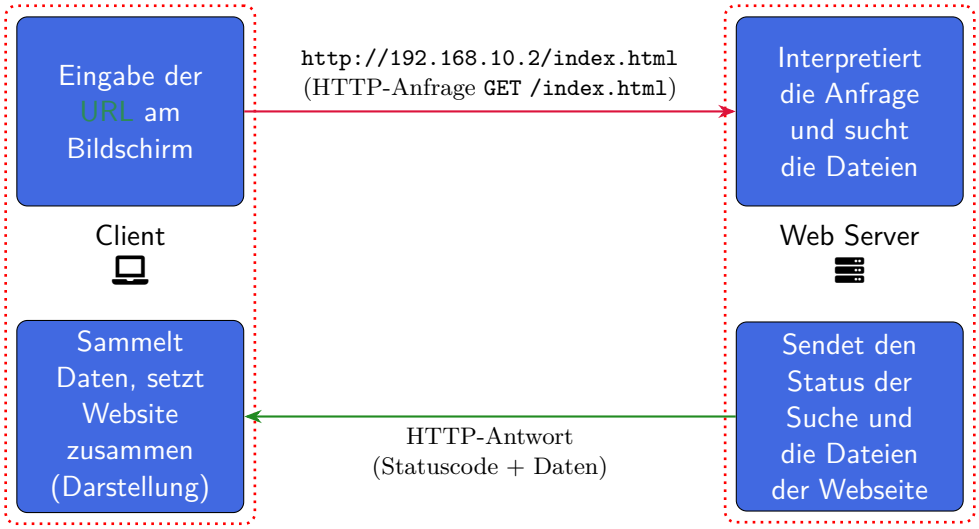



Abbildung 1.5: Schematischer Ablauf eines Webseiten-Aufrufs mit dem Hypertext Transfer Protocol (HTTP)-Protokoll

ig:http-request-simple

Wir können den Ablauf einer Abfrage etwas detaillierter aufzeigen, indem wir eine weitere Stufe für den End-Benutzer des Clients einfügen. Hierdurch wird ersichtlich, dass der Browser sich um die Kommunikation mit dem Server kümmert, indem er das HTTP-Protokoll verwendet, um Datei-Anfragen an den Server zu schicken und um diese zu erhalten (s. **??**).

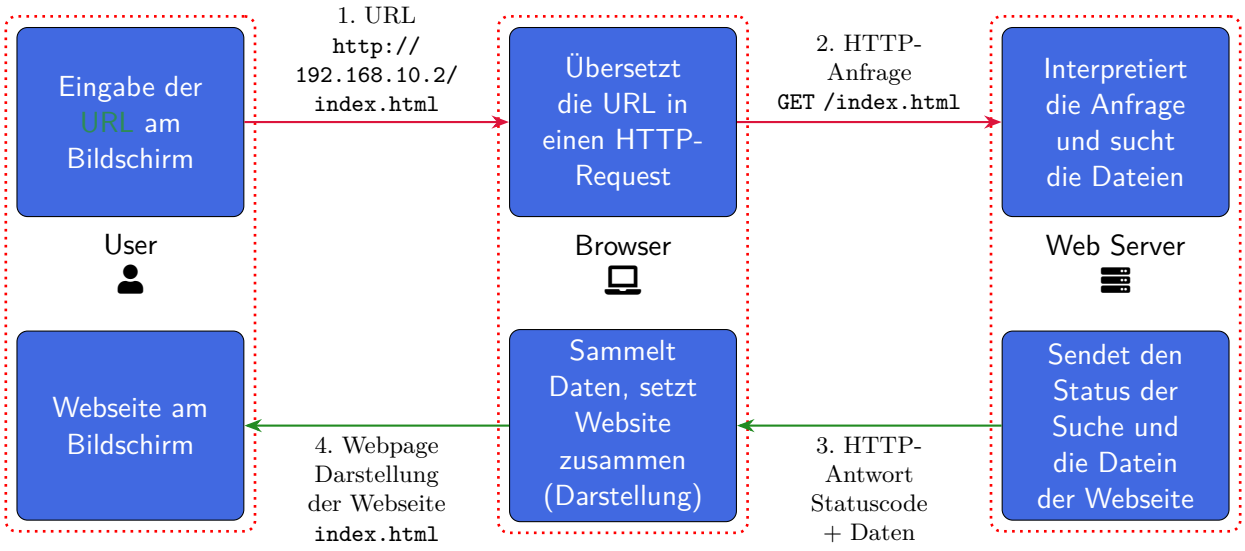



Abbildung 1.6: Schematischer Ablauf eines Webseiten-Aufrufs mit dem HTTP-Protokoll

ig:http-request

Damit der User die Nachricht jedoch tatsächlich erhält, müssen noch einige weitere Schritte erfolgen. Wie werden beispielsweise die Daten vom Server zum Client transportiert? Dafür müssen wir uns näher mit der Transportschicht auseinandersetzen.

**↑ Aufgabe (Abgabe) 1.15 Das Internet**

Schauen Sie sich jeweils zuerst die Videos an und erledigen Sie danach Aufgaben 2.7-2.9 auf folgendem Link. Laden Sie Ihre Lösungen als `.fls`-Dateien auf Moodle hoch.

**Aufgabe (Challenge) 1.16**

Richten Sie sich Ihren eigenen Email-Server ein, indem Sie folgende Aufgaben ausführen: Link.

#### 1.2.5.6 DNS

Wie wir bereits gesehen haben, besteht eine URL, also eine Adresse einer Webseiten, aus der Angabe eines Servers und einem Dateipfad. Diese Datei fordert der Client von einem Server mittels dem HTTP- oder Hypertext Transfer Protocol Secure (HTTPS)-Protokoll an (s. **??**). Allerdings möchte man sich nicht für jeden Server die IP-Adresse merken müssen: ähnlich Ihrer Kontakteliste auf dem Mobiltelefon, welche dazu dient, dass Sie sich nicht alle Telefonnummern auswending merken müssen, ist eine Domain Name System (DNS) essentiell eine Liste von merkbaren Webseiten-Namen (wie beispielsweise `sbb.ch`) und deren dazugehörigen IP-Adressen, also den IP-Adressen der Server, die die Webseiten-Datein speichern. Ein Beispiel einer solchen Tabelle ist in **??** gezeigt.

| Domain Name | IP-Addresse |
|---|---|
| `beispiel.ch` | 192.168.0.11 |
| `schule.ch` | 192.168.0.12 |
| `test.org` | 192.168.0.13 |
| `insta.com` | 192.168.0.14 |
| ... | ... |

Tabelle 1.5: Beispiel einer DNS-Tabelle
ab:dnsEntries

Wenn eine Adresse wie `insta.org` im Browser eingetippt und angefordert wird, schickt der Client eine Anfrage an einen nahegelegenen DNS-Server, der die IP-Adresse retourniert (ähnlich einem Telefonbuch). Ein DNS ist also ein Server, welcher den Clients eines Netzwerks eine Tabelle von Servern und damit verbundenen Namen zur Verfügung stellt. Eine solche DNS-Abfrage ist illustriert in **??**.

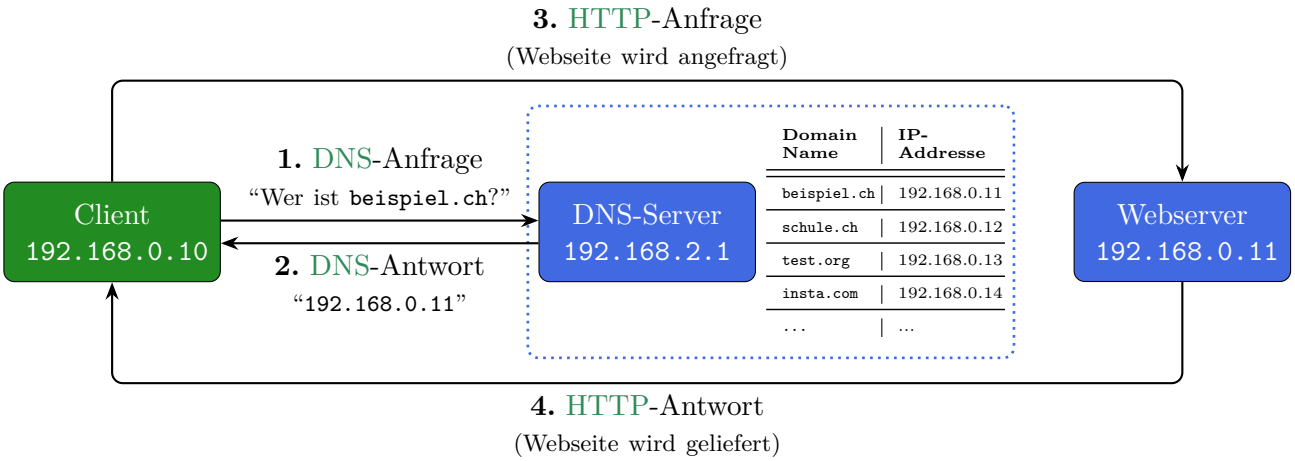



Abbildung 1.7: Schematischer Ablauf einer DNS-Anfrage
ig:dns

Der Prozess der Beantwortung einer Frage nach der IP-Adresse einer Web-Adresse ist in Schritten 1 und 2 in **??** gezeigt. Die Übersetzung eines Domain-Namens in eine IP-Adresse wird auch als **Namensauflösung** bezeichnet.

**Aufgabe 1.17**

Weshalb kann der Client `192.168.0.10` bei **??** nicht auf die Webseite `ksimlee.ch` zugreifen?

**Lösungsvorschlag zu Aufgabe 1.17**

Die Webseite `ksimlee.ch` ist nicht in der Tabelle des DNS-Servers `192.168.2.1` abgespeichert, daher kann sie der Client auch nicht finden.

Mit dem Terminal-Befehl `host [ipadress]` kann nachgeschaut werden, welche IP-Adresse zu einer bestimmten Webseite gehört. Mit dem Befehl `nslookup [ipadress]` erhalten Sie dieselben Informationen und zusätzlich dazu noch die IP-Adresse des DNS-Servers, der Ihnen die Informationen geliefert hat.

**Aufgabe 1.18**

Verwenden Sie die Befehle `host` oder `nslookup`, um herauszufinden, welche IP-Adresse zur Webseite `sbb.ch` gehört. Geben Sie diese IP im URL-Fenster Ihres Browser ein und beobachten Sie, was passiert.

**Lösungsvorschlag zu Aufgabe 1.18**

Die IP-Adresse von `sbb.ch` kann mit dem Befehl `host sbb.ch` ermittelt werden und ist `194.150.245.142`. Wenn wir die Adresse im Browser eingeben, landen wir auf der Seite der SBB.

**↑ Aufgabe (Abgabe) 1.19 DNS-Server**

Schauen Sie sich jeweils zuerst die Videos an und erledigen Sie danach Aufgaben 2.10-2.12 auf folgendem Link. Laden Sie Ihre Lösungen als `.fls`-Dateien auf Moodle hoch.

### 1.2.6 Vermittlungsschicht

#### 1.2.6.1 MAC-Adressen

Bisher haben wir mehrheitlich darüber gesprochen, wie Datenpakete zwischen adressierten Empfängern (s. IP-Adressen) hin- und hergeschickt werden, und wie Addressen in (Sub-)Netzwerke unterteilt und in verständliche Namen umgewandelt werden können. Eine IP-Addresse kann mit der Adresse eines Hauses verglichen werden. Allerdings legt der Briefträger die Briefe und Pakete nicht an einer Adresse ab, sondern in einem Briefkasten, also einem konkreten Objekt. Falls sich der Briefkasten oder sogar das ganze Haus ändert, so bleibt die Adresse doch dieselbe. Ähnlich dieser Analogie kann in einem Computer die gesamte Hardware ausgetauscht werden, inklusive der Netzwerkkarte, welche sich um das Verschicken und Empfangen von Daten kümmert, und trotzdem sollte der Datentransport weiterhin funktionieren. Damit dies klappt, benötigen wir ein weiteres Protokoll, welches die IP-Adresse in eine so genannte Media Access Control (MAC)-Adresse umwandelt. Die MAC-Adresse ist die physische Adresse der Netzwerkkarte in einem Computer, also sozusagen der "Fingerabdruck" der Netzwerkkarte. Sie wird normalerweise mit 48 bit angegeben, welche normalerweise der Kürze wegen in hexadezimaler Schreibweise notiert werden. Ein Beispiel einer MAC-Adresse könnte wie folgt aussehen: `48-2C-6A-1E-59-3D`

#### 1.2.6.2 ARP

Damit eine IP-Adresse in eine MAC-Adresse umgewandelt werden kann, wird ein weiteres Protokoll benötigt, das Address Resolution Protocol (ARP).

Ein Beispiel, wie das ARP-Protokoll funktioniert, könnte wie folgt aussehen:

- Computer `192.168.0.33` will Nachricht senden an `192.168.0.34` im selben Subnetz (`255.255.255.0`)
- `192.168.0.33` sendet daher eine Nachricht an alle Computer in `192.168.0.x`: "Wer hat die IP `192.168.0.34`"?
- Computer `192.168.0.34` antwortet: "Ich! Meine MAC ist `60:4E:46:F5:08:09`."
- Nun kann der Computer seine Nachricht direkt an den nächsten Empfänger (*gateway*) übergeben.

**Aufgabe 1.20 Weitere Filius-Aufgaben (Troubleshooting)**

Lösen Sie die Troubleshooting-Aufgaben 2.13 auf folgendem Link.

**Aufgabe (Challenge) 1.21**

- Lesen Sie zur Vertiefung des Schichtenmodells folgende Webseite (ohne Übungen): Link
- Beantworten Sie danach die Zuordnungs-Fragen → **Abgabe Moodle**

# Anhänge

## G Installation Filius

### G.1 Windows

Laden Sie folgende Datei herunter und installieren Sie das Programm: Link.

Falls Sie dazu aufgefordert werden, klicken Sie immer auf "erlauben", bzw. "weiter". Nachdem Sie das Programm installiert haben, können Sie folgende Schritte ausführen:

1. Das heruntergeladene Installationsprogramm (mit der Dateiendung `.exe`) löschen
2. Auf der Tastatur tippen: ⊞+`Filius`
3. Filius öffnen

### G.2 MacOS

**Achtung**

Falls Probleme beim Ausführen der unten stehenden Schritte auftreten, schauen Sie sich die genauen Anleitungen mit Screenshots auf dieser Webseite an: https://oinf.ch/wp-content/uploads/Anleitung_Filius_Mac-OS.pdf. Sie können natürlich auch mich um Hilfe fragen 😉.

**Installation Java SDK (Voraussetzung für Filius)** Damit Filius ausgeführt werden kann, muss zuerst Java SDK in der neusten Version installiert werden.

1. Laden Sie Java als DMG-Installer unter folgendem Link herunter: https://www.oracle.com/java/technologies/downloads/
   - Überprüfen Sie, ob Sie über einen Mac mit Apple- oder Intel-Chip verfügen, indem Sie ganz oben links auf Ihrem Bildschirm auf das -Zeichen → "Über diesen Mac" klicken und die Beschreibung Ihres Chips (erste Zeile) lesen.
   - Falls Sie einen neueren Mac mit Apple-Silicon-Chip (z.B. M1, M2, M3...) haben (Modelle nach 2022 gekauft), laden Sie die "ARM"-Version herunter (als `.dmg`-Datei)
   - Falls Sie einen Mac mit Intel-Chip haben, laden Sie die "x64"-Version herunter

Installieren Sie Java SDK, indem Sie auf die heruntergeladene Datei klicken.

**Installation Filius**

- Laden Sie Filius in der neusten Version **als `.zip`-Datei** von folgendem Link herunter: https://www.lernsoftware-filius.de/Herunterladen.
- Verschieben (ziehen) Sie den Ordner in Ihren Anwendungs-Ordner.
- Öffnen Sie den Ordner und starten Sie die Datei `filius.jar`, indem Sie rechts darauf klicken (⌥ + Klick) und "Öffnen" wählen. Dies müssen Sie nur beim ersten Mal machen, danach können Sie "normal" auf das Programm klicken (ohne ⌥), um es zu öffnen.

# Glossar

**ARP** Address Resolution Protocol. 17

**DNS** Domain Name System. 6, 15, 16

**FTP** File Transfer Protocol. 6

**HTTP** Hypertext Transfer Protocol. 6, 14–16
**HTTPS** Hypertext Transfer Protocol Secure. 6, 15

**IMAP** Internet Message Access Protocol. 6
**IP** Internet Protocol. 4, 8, 10–17

**MAC** Media Access Control. 17

**OSI** Open Systems Interconnection. 4

**POP3** Post Office Protocol Version 3. 6

**SMTP** Simple Mail Transfer Protocol. 6
**SSH** Secure Shell. 6
**SSL** Secure Sockets Layer. 6

**TCP** Transmission Control Protocol. 4, 6, 7
**TLD** Top-Level Domain. 13, 14

**URL** Uniform Resource Locator. 13–15