Cecilien-Gymnasium Düsseldorf

Rechnernetze

Arbeitsheft für Schülerinnen und Schüler

Dr. Daniel W. Appel

Düsseldorf, 12. September 2016

Inhaltsverzeichnis

1 IP-Adressen

Jeder an das Internet angeschlossene Rechner benötigt eine Adresse, damit er identifiziert werden kann und Daten an ihn geschickt werden können. Diese Adresse ist seine sogenannte IP-Adresse[1].

Eine IP-Adresse ist nicht für alle Zeiten fest mit einem Rechner verbunden. Sie wird ihm, wenn er sich beispielsweise in einem Heimnetzwerk anmeldet, automatisch zugeteilt. Neugierige können einmal auf die Seite `http://www.meine-aktuelle-ip.de` gehen und sich die aktuelle IP-Adresse ihres Rechners oder Handys anzeigen lassen.

In dem heute noch gebräuchlichen IPv4-Standard besteht eine IP-Adresse aus vier Zahlen von 0 bis 255, also zum Beispiel:

76.134.33.64

Technisch etwas genauer gesagt, besteht eine IP-Adresse aus vier Bytes, denn mit einem Byte (= 8 Bit) lassen sich genau die Zahlen von 0 bis 255 in der Binärschreibweise darstellen. Die oben genannte Adresse würde man in der Binärschreibweise als

01001100.10000110.00100001.01000000

schreiben, was natürlich deutlich schwerer zu lesen ist.

Als Privatnutzer müssen wir uns über die Zuteilung einer IP-Adresse im Grunde keine Gedanken machen. Wären wir Administrator eines großen Unternehmens sähe das allerdings vielleicht anders aus. Unternehmen oder auch öffentliche Einrichtungen wie Universitäten erhalten meist Adressblöcke zugeteilt. D.h. sie erhalten einen Vorrat an IP-Adressen, die sie dann an ihre Endgeräte verteilen dürfen. Ist der Bereich zu klein, wird es problematisch.

Eine IP-Adresse wird stets aufgeteilt in einen *Netzwerkteil* und einen *Geräteteil*. Beispielsweise könnte das Gerät mit der oben genannten IP-Adresse in dem Netzwerk mit der Adresse 76.134.33 stehen und innerhalb dieses Netzwerkes den letzten Adressteil 64 zugeteilt bekommen haben. Der Netzwerkteil wäre also 76.134.33 und der Geräteteil 64. Wie die Unterteilung in diese beiden Teile funktioniert, sollten wir uns aber noch etwas genauer ansehen, denn leider ist die Trennung in die beiden Teile nicht immer so einfach.

Die Aufteilung in Netzwerk- und Geräteteil geschieht mithilfe der sogenannten *Subnetzmaske*. Die Subnetzmaske ist eine 32-Bit Binärzahl und hat damit genau so viele Stellen wie eine IP-Adresse. An den Stellen, die zum Netzwerkteil gehören steht in der Subnetzmaske eine 1 und an den übrigen Stellen eine 0. In unserem Beispiel hätten wir also folgende Situation:

IP-Adresse	01001100.10000110.00100001.01000000	=	76.134.33.64
Subnetzmaske	11111111.11111111.11111111.00000000	=	255.255.255.0
Netzwerkteil	01001100.10000110.00100001.00000000	=	76.134.33.0
Geräteteil	00000000.00000000.00000000.01000000	=	0.0.0.64

Für ein anderes Gerät im selben Netz könnte die Situation vielleicht so aussehen:

IP-Adresse	01001100.10000110.00100001.00010010	=	76.134.33.17
Subnetzmaske	11111111.11111111.11111111.00000000	=	255.255.255.0
Netzwerkteil	01001100.10000110.00100001.00000000	=	76.134.33.0
Geräteteil	00000000.00000000.00000000.00010010	=	0.0.0.17

Eine **andere Subnetzmaske** würde zu einer neuen Situation führen. Zum Beispiel:

IP-Adresse	01001100.10000110.00100001.01000000	=	76.134.33.64
Subnetzmaske	11111111.11111111.11110000.00000000	=	255.255.240.0
Netzwerkteil	01001100.10000110.00100000.00000000	=	76.134.32.0
Geräteteil	00000000.00000000.00000001.01000000	=	0.0.1.64

Oder auch:

IP-Adresse	01001100.10000110.00100011.00010010	=	76.134.35.17
Subnetzmaske	11111111.11111111.11110000.00000000	=	255.255.240.0
Netzwerkteil	01001100.10000110.00100000.00000000	=	76.134.32.0
Geräteteil	00000000.00000000.00000011.00010010	=	0.0.3.17

Einfach gesagt, gibt die Subnetzmaske also an, wie viele Stellen in der Binärschreibweise für den Netzwerkteil verwendet werden und wie viele für den Geräteteil übrig bleiben. In der Dezimalschreibweise ist diese Unterteilung etwas verschleiert.

Bleiben wir einmal bei der zweiten Subnetzmaske 255.255.240.0 mit den insgesamt 20 Einsen. Angenommen, wir wären Besitzer des Netzwerks, in dem sich eines der Geräte aus den Beispielen befindet. Welche IP-Adressen stünden uns dann zur Vergabe in unserem Netzwerk zur Verfügung? Das können wir uns vielleicht klar machen, wenn wir uns ansehen, wie viel Spielraum wir bei der Vergabe der IP-Adressen in unserem Netzwerk haben:

[1] IP steht für *Internet Protocol*.

Subnetzmaske	11111111.11111111.11110000.00000000	=	255.255.240.0
kleinste mögliche IP-Adresse	01001100.10000110.00100000.00000000	=	76.134.32.0
kleinster möglicher Geräteteil	00000000.00000000.00000000.00000000	=	0.0.0.0
größte mögliche IP-Adresse	01001100.10000110.00101111.11111111	=	76.134.47.255
größter möglicher Geräteteil	00000000.00000000.00001111.11111111	=	0.0.15.255
Netzwerkteil	01001100.10000110.00100000.00000000	=	76.134.32.0

Letztlich können wir also die hinteren 12 Bits der IP-Adresse frei variieren. Das liefert uns $2^{12} = 4096$ IP-Adressen, die wir vergeben können. Für viele Unternehmen wäre so ein Adressblock sicherlich ausreichend. Zwei Einschränkungen gibt es allerdings noch:

- Die kleinste mögliche IP-Adresse ist als Bezeichnung für das Netzwerk selbst reserviert. D.h. unser Beispielnetzwerk ist das Netzwerk mit der IP-Adresse 76.134.32.0.
- Die größte mögliche IP-Adresse ist für einen Broadcast, d.h. für ein versenden von Daten an alle Rechner im Netzwerk, reserviert.

Man kann den Adressblock unseres Netzwerks übrigens auch kurz als

76.134.32.0/20

schreiben. D.h. man kann die Netzwerkadresse zusammen mit der Anzahl der Einsen in der Subnetzmaske angeben. Der Rest lässt sich daraus erschließen.

Aufgabe 1. Wir bleiben weiterhin bei unserem Netzwerk mit dem Adressblock 76.134.32.0/20.

(a) Ein Gerät hat in Binärschreibweise die IP-Adresse 01001100.10000110.00100011.01000100.
Gib den Geräteteil in Binärschreibweise an. Danach berechne die Dezimalschreibweise des Geräteteils.

(b) Ein Gerät hat in Dezimalschreibweise die IP-Adresse 76.134.40.120.
Ermittle den Geräteteil in Binär- und Dezimalschreibweise.

Aufgabe 2. Die deutsche Telekom hat den Adressblock 217.224.0.0/11 erhalten[2].

(a) *Gib die Subnetzmaske der Telekom in Binär- und Dezimalschreibweise an.*

(b) *Ermittle die größtmögliche IP-Adresse, die die Telekom verwenden kann, in Binär- und Dezimalschreibweise.*

Aufgabe 3. (a) In der Fachliteratur liest man oft, dass sich der Netzwerkteil aus der IP-Adresse und der Subnetzmaske mittels des logischen UND ergibt.
Erkläre, wie dies zu verstehen ist.

(b) *Beschreibe, wie man aus IP-Adresse und Netzwerkteil in Dezimalschreibweise den Geräteteil (einfach) ermitteln kann. Erkläre auch, warum das von Dir beschriebene Vorgehen funktioniert!*

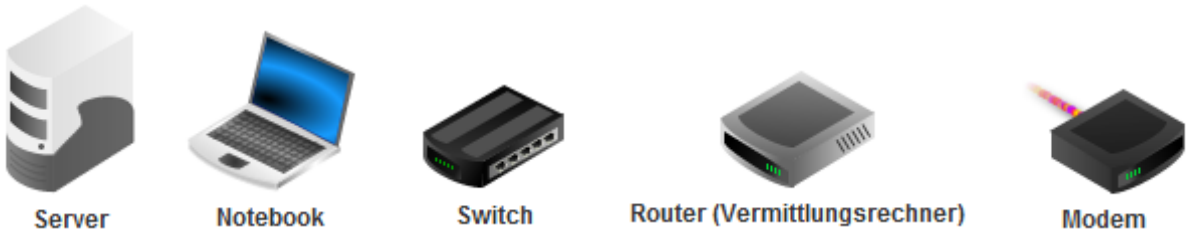
[2]Dieses ist *Einführung in die Informatik* von H.P. Grumm und M. Sommer entnommen

2 Erste Schritte in FILIUS

Um uns die Konstruktion von verschiedenen Netzwerken und deren Arbeitsweisen zu veranschaulichen, verwenden wir die Lernsoftware FILIUS[3]. In den folgenden Abschnitten sehen wir uns erst einmal die grundlegenden Funktionen an.

2.1 Hardwarekomponenten

Die folgenden Hardwarekomponenten stehen uns in FILIUS zur Verfügung.



Server: Ein Server ist letztlich ein gewöhnlicher Rechner, der spezielle Dienste zur Verfügung stellt. Beispielsweise kann auf ihm ein Emaildienst angeboten werden. In der Praxis sind dies natürlich meist sehr leistungsstarke Rechner und keine gewöhnlichen Heimrechner.

Notebook: Für uns steht ein Notebook einfach stellvertretend für ein Gerät mit dem ein Benutzer einen Netzdienst verwenden möchte. In der realen Welt könnte es nicht nur ein Notebook, sondern auch ein gewöhnlicher Rechner, ein Tablet oder natürlich ein Smartphone sein. Für uns stellt ein Notebook damit stets einen *Client* dar — das ist grob gesprochen das Gegenstück zu einem Server.

Switch: Möchten wir mehr als nur zwei Rechner miteinander verbinden, benötigen wir einen zentralen Knotenpunkt. Ein Switch ist der einfachste, den wir dazu wählen können. Ein Switch kann nur Rechner innerhalb eines Netzwerks miteinander verbinden.

Router: Mit einem Router bzw. Vermittlungsrechner können Netzwerksignale von einem Netzwerk in ein anderes versendet werden. Dazu muss er aber zunächst manuell eingerichtet werden.

Modem: Mit einem Modem können über ein reales Netzwerk mehre Filius-Programme miteinander vernetzt werden. Darauf gehen wir hier jedoch nicht weiter ein.

In vielen Haushalten steht ein Gerät, dass mehrere der oben genannten Komponenten in sich vereint, so dass die Aufgaben der einzelnen Komponenten uns aus dem Alltag nicht vertraut sind oder die Unterscheidung dieser Aufgaben uns vielleicht überraschend erscheint. Es ist zum Beispiel sehr gut möglich, dass bei Euch zu Hause ein DSL-Modem mit mehreren Netzwerkanschlüsse steht, so dass man mehrere Rechner über Kabel anschließen kann. Es übernimmt damit also auch die Aufgabe eines Switches.

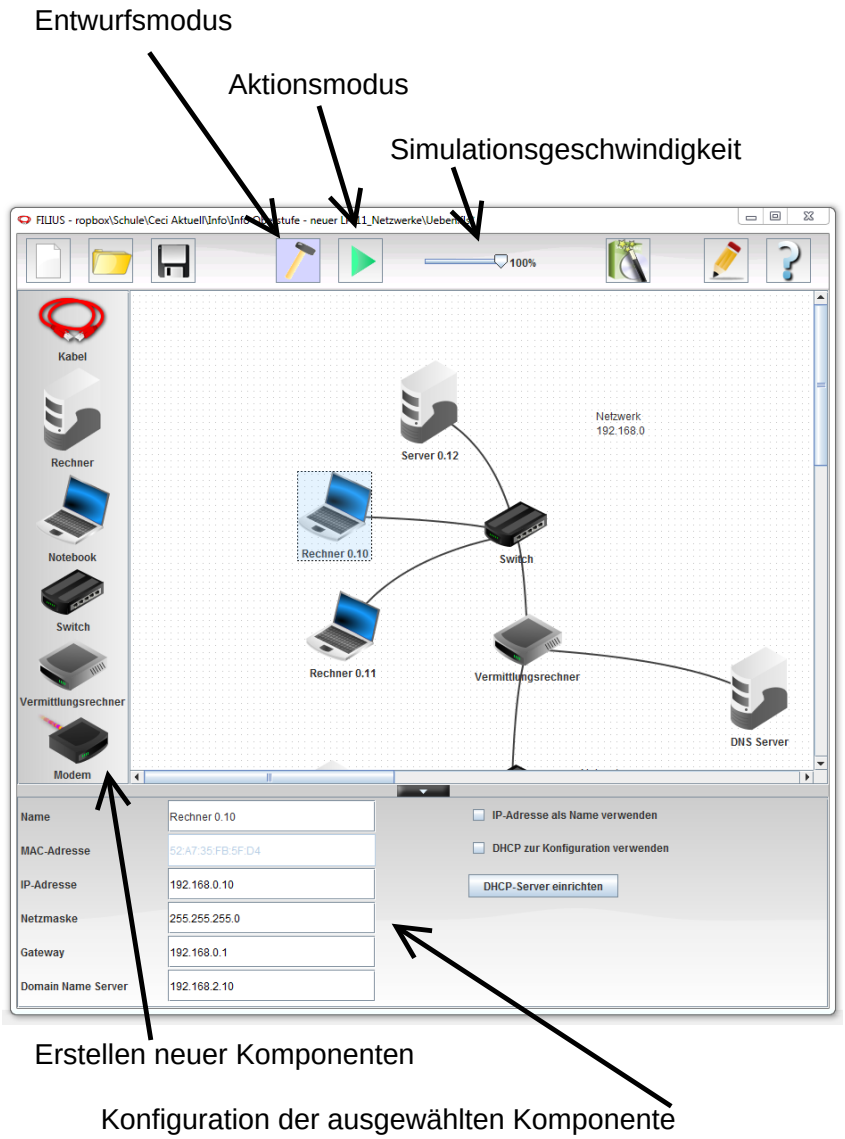
Eine weitere Komponente, die wir zum Aufbau eines Netzwerkes benötigen ist diese

auch, wenn sie im Vergleich zu den oben beschriebenen natürlich ein wenig aus dem Rahmen fällt. Übrigens stellt so ein Kabel in Wahrheit einfach eine Netzwerkverbindung zwischen zwei Geräten dar. Das soll bedeuten, dass es durchaus auch stellvertretend für eine Funkverbindung (WLAN!) stehen kann.

2.2 Das Arbeitsfenster von FILIUS

Das Programm FILIUS ist recht übersichtlich:

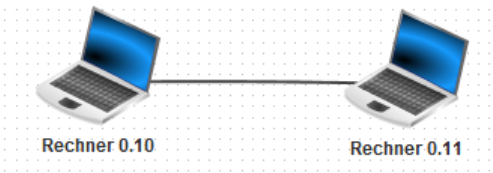
[3] `http://www.lernsoftware-filius.de/`



Zunächst können wir im **Entwurfsmodus** ein Netzwerk konstruieren. Dabei können wir links die Komponenten wählen, die wir verbauen wollen und diese dann im Menü unten konfigurieren. Um unser Netzwerk zu testen, müssen wir in den **Aktionsmodus** wechseln.

2.3 Verbinden zweier Rechner

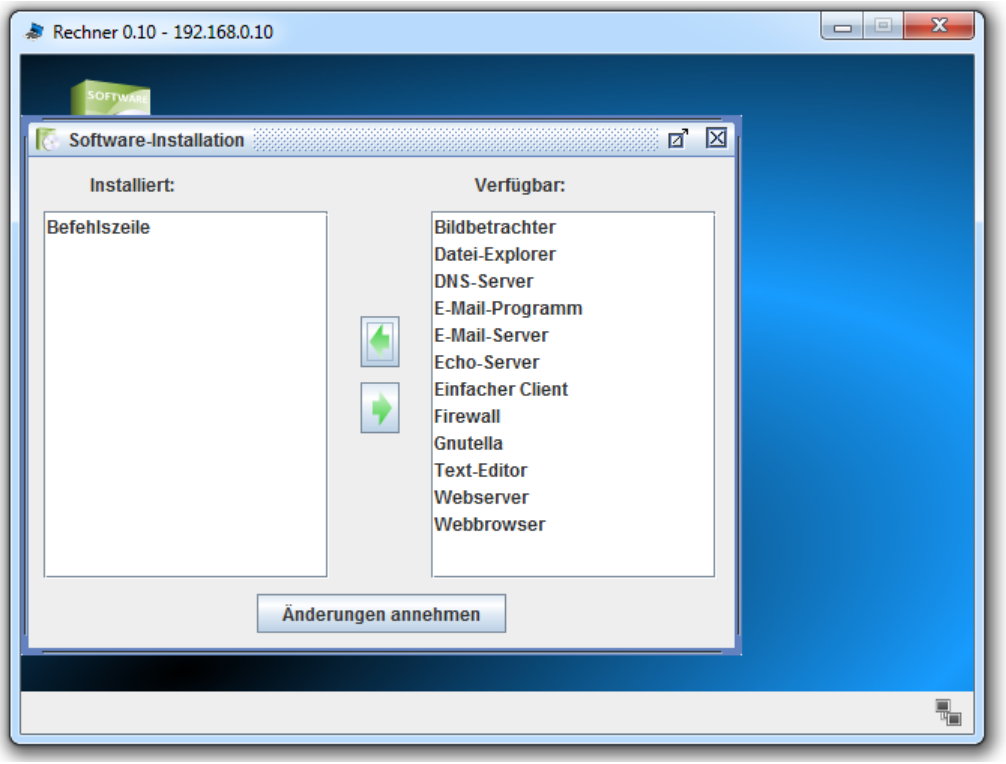
In unserem ersten Beispiel wollen wir einfach zwei Rechner direkt miteinander verbinden:



Die beiden Rechner sollen die IP-Adressen 192.168.0.10 und 192.168.0.11 besitzen. Als Subnetzmaske verwenden wir 255.255.255.0, so dass sie sich beide im selben Netzwerk befinden.

Um zu testen, ob dieses minimalistische Netzwerk funktioniert, müssen wir in den Aktionsmodus wechseln. Dort klicken wir auf den Rechner mit IP 192.168.0.10 und öffnen so seinen Desktop. Auf diesem befindet sich derzeit nur ein einziges Symbol zur Installation von Software:

Dieses klicken wir wiederum an, um die Anwendung *Befehlszeile* zu installieren:



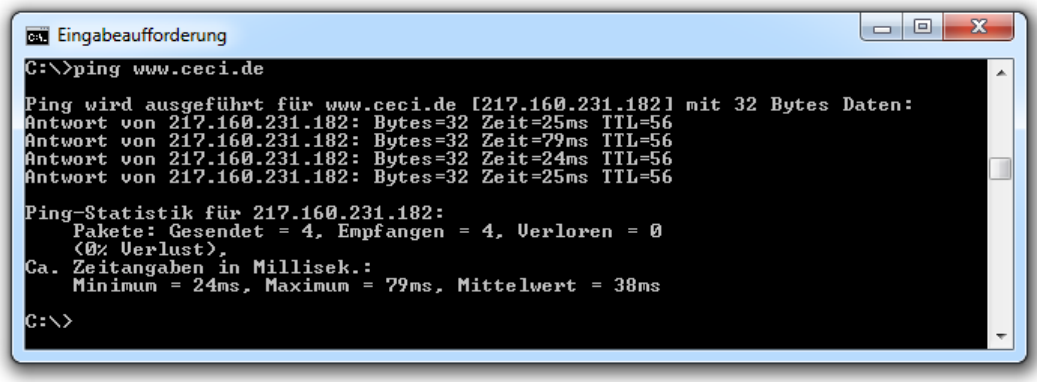
Nehmen wir die Änderungen an, erscheint auf dem Desktop das entsprechende Symbol.

Nun können wir durch Klick auf dieses Symbol die Kommandozeile öffnen, um dort Systembefehle einzugeben. Heutzutage erscheint die Arbeit mit einer solchen Kommandozeile sicherlich etwas altmodisch — wenn nicht sogar vollkommen fremd, denn die Zeiten, in denen ein Rechner nach dem Start den Benutzer mit einer solchen Kommandozeile begrüßte sind lange vorbei — für Netzwerktechniker gehört die Arbeit mit ihr aber zum täglichen Brot[4].

Geben wir den Befehl `ipconfig` ein, so erhalten wir Informationen über die Netzwerkkonfigurationen des Rechners.

Mit dem Befehl `ping 192.168.0.11` können wir die Verbindung zum zweiten Rechner testen. Dazu werden probeweise vier Datenpakete an ihn gesendet und eine Bestätigung über den Empfang erwartet. Ist die Verbindung in Ordnung, sollte angezeigt werden, dass alle Pakete gut angekommen sind.

Diese beiden Befehle gibt es übrigens durchaus auch in der Praxis! Wir können in Windows die Eingabeaufforderung starten und dort beispielsweise testen, ob der Server der Ceci-Homepage erreichbar ist:

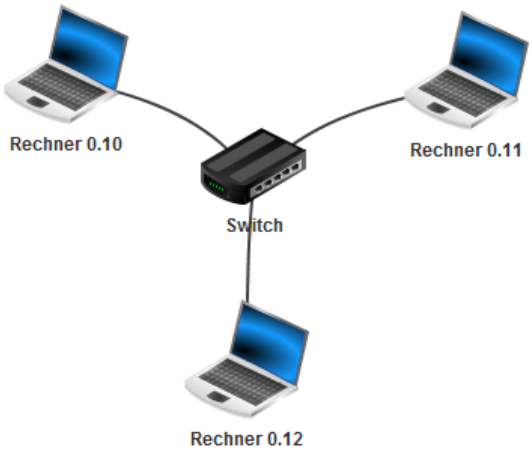


[4]Vor **Windows 95** war das gängigste Betriebssystem **MS-DOS**. Dieses arbeitete mit einer solchen Kommandozeile. Mittels geeigneter Kommandos konnte man sich in den Verzeichnissen der Festplatte bewegen, Dateien kopieren und löschen oder natürlich auch Programme starten. Nutzer von **Linux** tendieren auch heute noch dazu, die Kommandozeile wesentlich intensiver zu nutzen als Windows- oder Mac-User.

Aufgabe 4. *Falls nicht ohnehin schon geschehen, vollziehe die oben beschriebene Verbindung zweier Rechner selbst mit FILIUS nach. Speichere diese Simulation ab!*

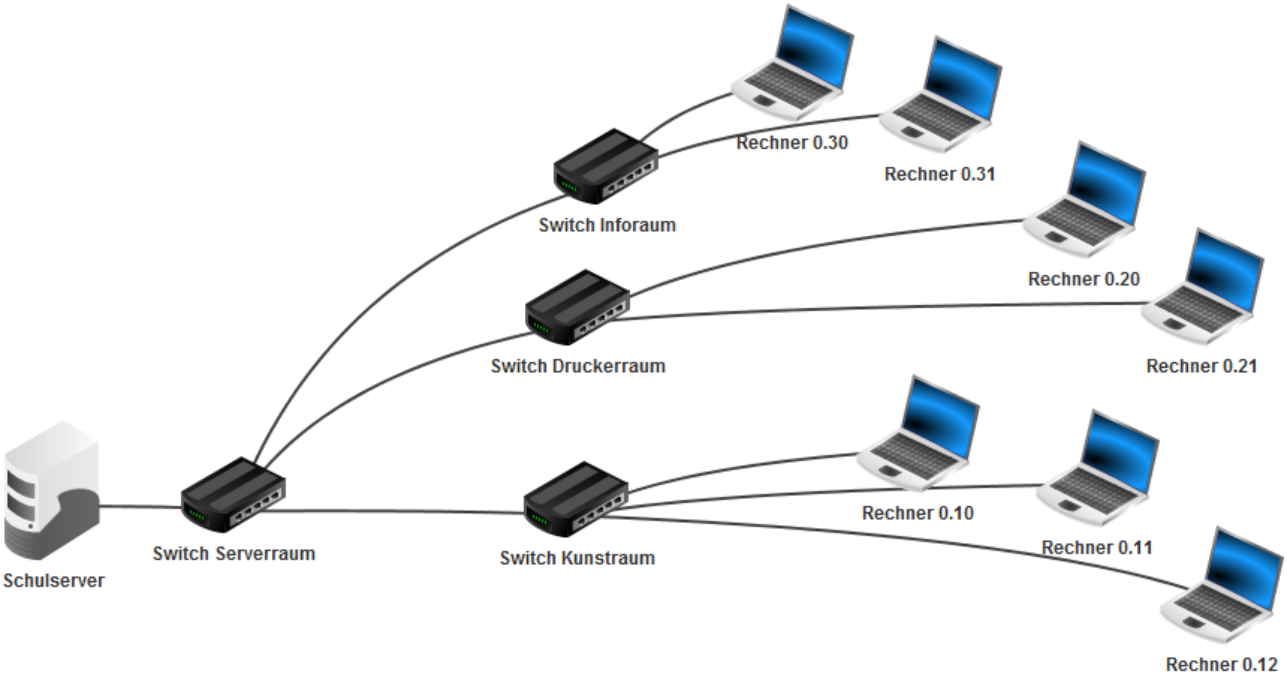
2.4 Verbinden mehrerer Rechner

Möchten wir mehr als zwei Rechner miteinander verbinden, können wir diese nicht mehr direkt untereinander verkabeln. Wir benötigen einen Knotenpunkt, der die Verteilung der Datenpakete an das für sie bestimmte Ziel übernimmt. Für diese Aufgabe ist ein Switch geeignet:



Aufgabe 5. *Entwickle ein Netzwerk mit drei Rechnern und teste die Verbindungen.*
Tipp: Verbinde zunächst die Komponenten so wie in der Abbildung. Die Netzwerkkonfigurationen laufen danach im Grunde so ab wie in Aufgabe 4 nur eben mit einem zusätzlichen Rechner mit einer weiteren IP-Adresse.

Die folgende Abbildung stellt eine vereinfachte Version unseres Schulnetzwerks dar. Wie wir sehen, gibt es hier vier Switches: Einen im Serverraum und einen weiteren in jeder Etage.



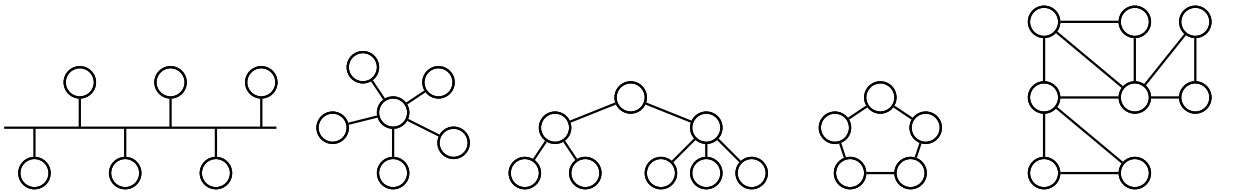
Im Serverraum steht natürlich auch noch unser Server, der wiederum mit dem Internet verbunden ist. Dies soll für den Moment noch keine Rolle spielen. Momentan ist der Server für uns auch einfach nur ein Rechner. Lediglich sein Symbol sieht anders aus.

Aufgabe 6. *Erstelle in FILIUS die hier dargestellte Version unseres Schulnetzes. Der Server soll dabei die IP-Adresse* 192.168.0.1 *erhalten. Teste mithilfe verschiedener* `ping`*-Anweisungen, ob die Rechner sich untereinander und auch den Schulserver finden.*

3 Netzwerktopologien

3.1 Einführung

Es gibt verschiedene sogenannte *Topologien*, die man für ein Netzwerk wählen kann. Die Topologie eines Netzwerks beschreibt, in welcher Struktur die Geräte miteinander verbunden werden. Was damit gemeint ist, wird vielleicht deutlich, wenn man sich einfache schematische Darstellungen der gängigsten Topologien ansieht:



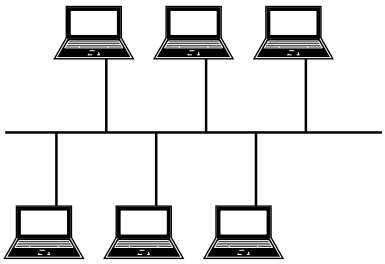
Bustopologie | Sterntopologie | Baumtopologie | Ringtopologie | Vermaschte Topologie

Wie oben schon erwähnt, geht es hier um die Struktur der Verbindungen. Vernachlässigt werden hier Aspekte wie zum Beispiel die Länge der Verbindungen oder ob es sich um Kabel- oder Funkverbindungen handelt.

Aufgabe 7. *Entscheide, welche der Topologien beim Netzwerk unserer Schule (siehe Aufgabe 6) vorliegt.*

3.2 Bustopologie

In der Bus-Topologie werden alle Geräte hintereinander mit einer gemeinsamen Leitung — dem Bus — verbunden. Es gibt hier keine zentrale Steuereinheit, die den Datenverkehr im Bus regelt. Das führt dazu, dass immer nur ein Gerät gleichzeitig senden darf, da es sonst zu Kollisionen im Datenverkehr kommen könnte.



Diese Topologie findet heute bei Netzwerken kaum noch Anwendung. Die Idee eines Datenbus wird aber durchaus an anderen Stellen noch verwendet! Die Komponenten eines Von-Neumann-Rechners sind mit einem Bus verbunden und dort ist es sogar gewollt, dass immer nur eine Komponente sendet. Auch beim Anschluss von USB-Geräten kommt diese Topologie zum Einsatz — USB steht schließlich für *Universial Serial Bus*.

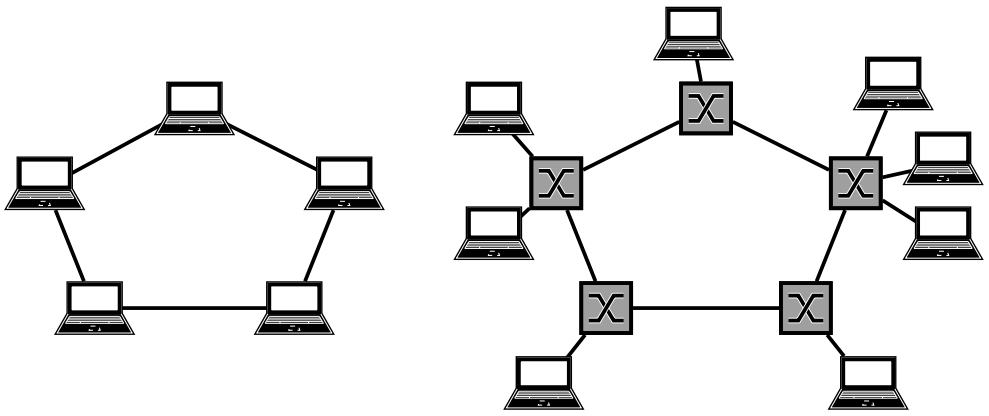
Sehen wir uns noch die einige Vor- und Nachteile dieser Topologie an:

+ Fällt ein Gerät aus, kann das restliche Netzwerk ohne Einschränkungen weiterarbeiten.
+ Da nur wenige (Kabel-)Verbindungen nötig sind, sind die Kosten gering.
+ Simple Verkabelung und einfache Netzerweiterung.
+ Aktive Netzwerkkomponenten werden nicht benötigt. Das hält ebenfalls die Kosten gering.

– Datenübertragungen können leicht abgehört werden, da alle durch denselben Kanal laufen.
– Schon eine Störung des Datenbus an einer einzigen Stelle blockiert das gesamte Netzwerk.
– Es kann zu jedem Zeitpunkt immer nur ein Gerät Daten senden. Währenddessen müssen alle anderen warten.

3.3 Ringtopologie

In der Literatur findet man meist zwei Umsetzungsmöglichkeiten für die Ringtopologie. Links in der Abbildung sehen wir die einfachere Variante, in der jeder Rechner mit einem Nachfolger und einem Vorgänger verbunden ist. Das Senden der Daten verläuft hier immer **nur in eine Richtung** durch Weitergabe bis die Daten beim korrekten Empfänger angekommen sind.



Eine etwas flexiblere Interpretation dieser Topologie besteht aus einem Ring von Verteilern (z.B. Switches), an denen dann jeweils ein oder mehrere Rechner angeschlossen sind. Das sehen wir in der Abbildung rechts. Diese Variante hat unter anderem den Vorteil, dass ein Ausfall eines Rechners das übrige Netzwerk nicht beeinträchtigt.

Weitere Vor- und Nachteile dieser Topologie:

+ Das Netzwerk kann sehr groß sein (im Sinne einer großen räumlichen Ausbreitung), weil immer nur zwei benachbarte Stationen miteinander verbunden werden müssen und jede Station als Signalverstärker dienen kann.

+ Leicht programmierbar.

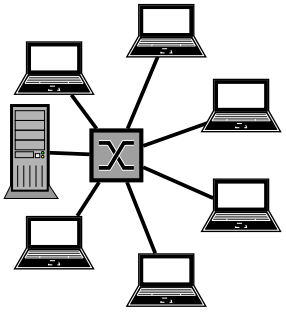
– Störung eines Gerätes oder der Leitung an einer Stelle kann zu Ausfall des Netzes führen.

– Übertragungen können leicht abgehört werden.

– Bei vielen Endgeräten wird die Übertragung langsamer.

3.4 Sterntopologie

In der Sterntopologie sind alle Rechner an einen Zentralrechner oder ein Vermittlungssystem angeschlossen:



Einige Vor- und Nachteile dieser Topologie:

+ Fällt ein Endgerät aus, kann das Netzwerk ungestört weiterarbeiten.

+ Leicht erweiterbar.

+ Leichte Fehlersuche.

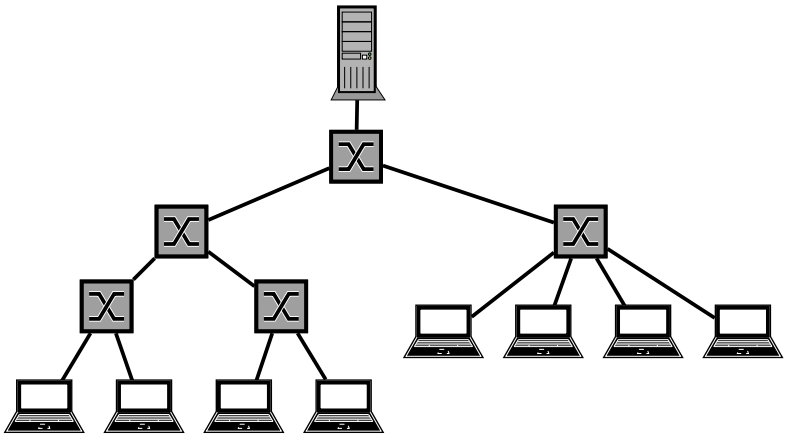
– Ausfall des gesamten Netzes bei Ausfall der Zentrale.

– Potentiell hohe Belastung der Zentrale.

– Verkabelung möglicherweise umständlich, da immer eine Verbindung zur Zentrale gelegt werden muss.

3.5 Baumtopologie

Die Baumtopologie kann man als Verallgemeinerung der Sterntopologie betrachten (denn ein Stern ist ein besonders einfacher Baum). An der Wurzel des Baumes und an allen inneren Knoten befindet sich ein jeweils Switch, wobei der Switch der Wurzel oft mit einem zentralen Server verbunden ist. Aufgrund der hohen Netzwerklast muss an der Wurzel ein besonders leistungsfähiger Switch verwendet werden.



Diese Topologie wird in den meisten Schulen oder anderen großen Gebäuden verwendet.

Einige Vor- und Nachteile dieser Topologie:

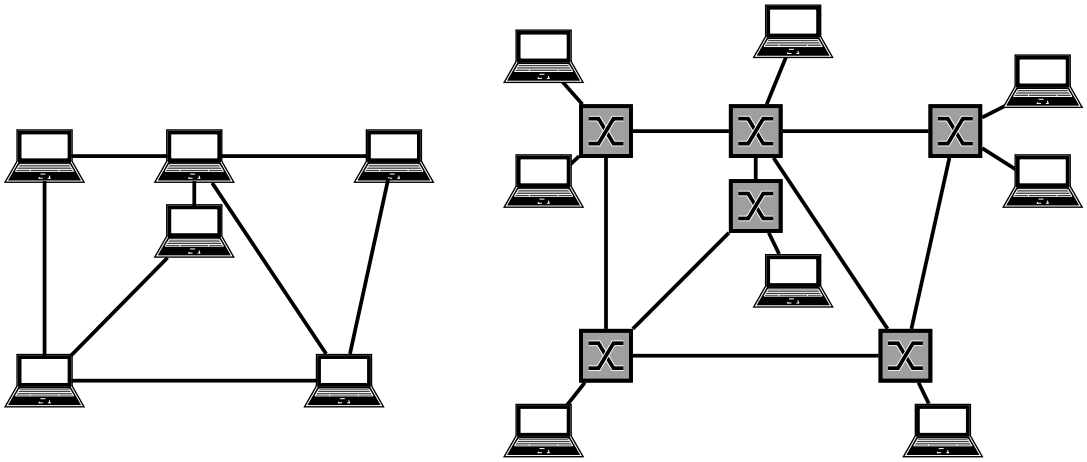
+ Fällt ein Endgerät aus, kann das Netzwerk ungestört weiterarbeiten.
+ Leicht erweiterbar.
+ Netzwerk kann sehr groß sein (im Sinne einer großen räumlichen Ausbreitung).

– Ausfall eines inneren Knotens führt zu Ausfall eines gesamten Teilnetzes.
– Hoher Datenverkehr an der Wurzel kann zu Verzögerungen führen.
– Bei tiefen Bäumen potentiell lange Übertragungswege.

3.6 Vermaschte Topologie

In einem vermaschten Netz gibt es im Gegensatz zu den bisher betrachteten im Allgemeinen mehrere mögliche Wege von einem Rechner zu einem anderen. Das führt dazu, dass selbst bei Ausfall eines Knotens das übrige Netz in den meisten Fällen weiterarbeiten kann — die Daten können dann umgeleitet werden.

Ähnlich wie bei der Ringtopologie findet man auch hier oft zwei Darstellungen:



Einige Vor- und Nachteile dieser Topologie:

+ Hohe Ausfallsicherheit.
+ Hohe Leistungsfähigkeit und gute Verteilung der Auslastung.

+ Keine zentrale Verwaltung.

– Hoher Energieverbrauch.

– Aufwändige Verkabelung.

– Alle beteiligten Geräte sind oft aktiv und sollten nach Möglichkeit eingeschaltet bleiben (vor allem bei der Interpretation in der Abbildung links).

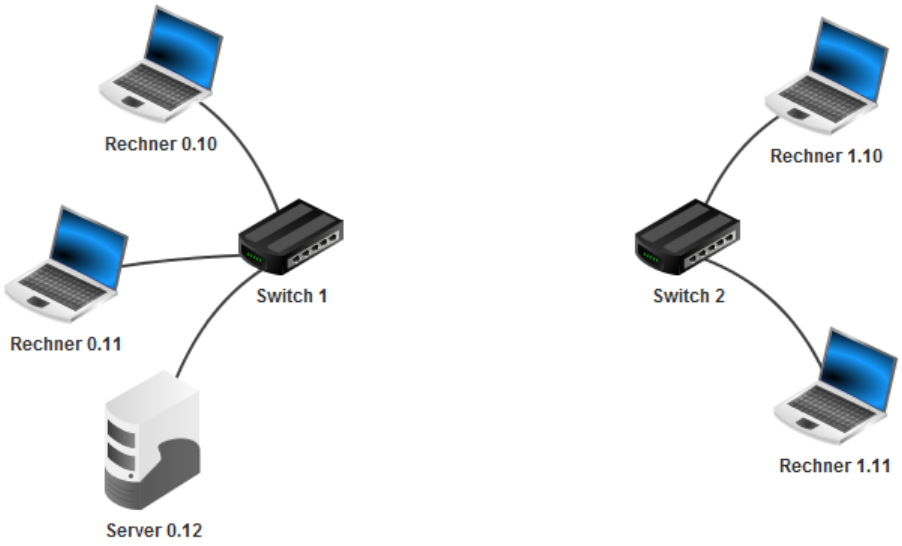
Diese Topologie kommt beim Internet zum Einsatz. Das Internet ist allerdings im Gegensatz zu den bisher von uns konstruierten Netzen ein Netzwerk von Netzwerken, so dass dort neben Rechnern und Switches noch weitere Komponenten benötigt werden. Wie man mehrere Netzwerke miteinander verbinden kann, sehen wir uns im folgenden Kapitel an.

4 Netze von Netzen

4.1 Verbinden zweier Netzwerke

Wir wollen uns nun ansehen, wie man Netzwerke miteinander verbinden kann und so Schritt für Schritt den grundlegenden Aufbau des Internets nachvollziehen.

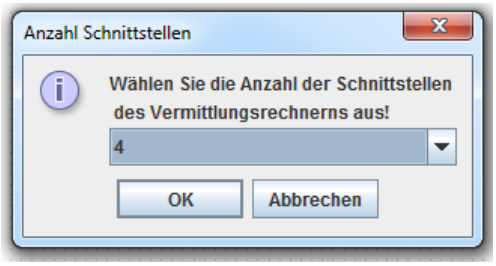
Aufgabe 8. *Erstelle zunächst zwei Netzwerke wie in der folgenden Abbildung. Als IP-Adressen solltest Du die Adressen* 192.168.0.10 *etc. verwenden und als Subnetzmaske die Standardeinstellung* 255.255.255.0.



Überprüfe mithilfe von `ping`*-Befehlen, ob die beiden einzelnen Netzwerke funktionieren.*

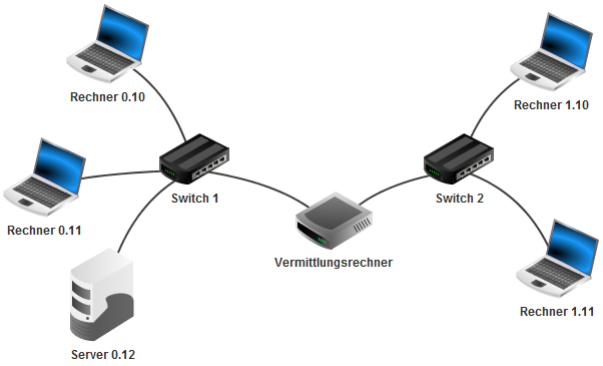
Nun müssen wir zwischen den beiden Netzwerken einen Router[5] hinzufügen.

Wenn wir den Router dort einfügen, werden wir gefragt, wie viele Schnittstellen er haben soll. Obwohl für den Moment zwei genügen, wählen wir vier Schnittstellen, um unser Modell später noch erweitern zu können. Der Router hat dann also vier Netzwerkkarten.



Schließlich sollte unser Modell so aussehen:

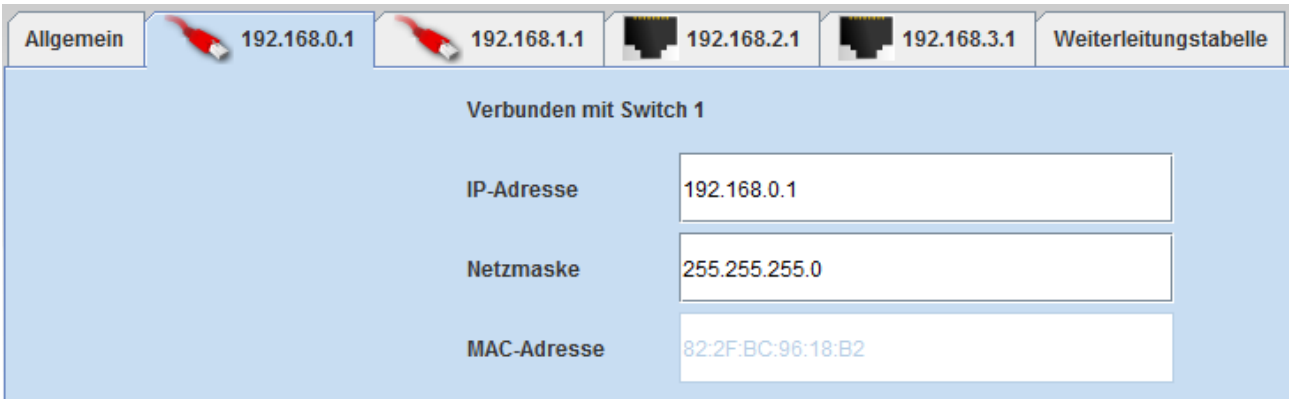
[5] In FILIUS wird diese Komponente *Vermittlungsrechner* genannt, weil die nicht den vollen Funktionsumfang eines Routers bietet. Wir können diese Unterscheidung aber zur Vereinfachung vernachlässigen.



Aufgabe 9. *Überprüfe mithilfe einer* `ping`*-Anweisung die Verbindung von Rechner* 0.10 *zu Rechner* 1.10. Hinweis: Nicht wundern, hier muss ein Fehler auftreten!

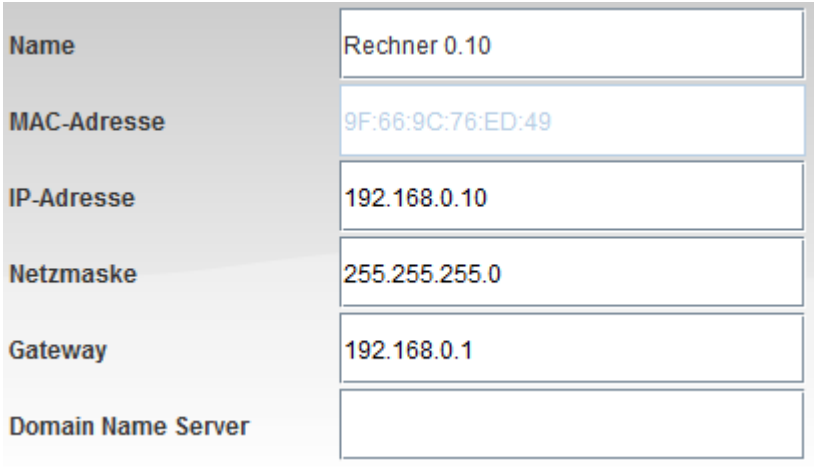
Die Anfrage von Rechner 0.10 müsste das Netzwerk des Rechners verlassen. Allerdings „weiß“ die Anfrage bisher noch nicht, wohin sie soll, wenn sie die gewünschte IP-Adresse nicht im Netzwerk vorfindet. Dieses Problem beheben wir mit einem *Gateway*.

Dazu müssen wir zunächst die IP-Adressen der im Router verbauten Netzwerkkarten kontrollieren. Im Entwurfsmodus öffnen wir durch Doppelklick auf den Router seine Konfigurationen. Wir sollten dort die in der Abbildung angegebenen IP-Adressen wählen. Dabei müssen wir darauf achten, dass die IP-Adresse 192.168.0.1 zum Anschluss des linken Netzwerkes gehört und 192.168.1.1 zum rechten. Welchen Anschluss wir gerade bearbeiten sehen wir daran, dass die angeschlossene Leitung grün leuchtet.



Achtung! Beachte, dass **alle vier** Netzwerkkarten von uns eine IP-Adresse zugeteilt bekommen, da wir Doppler bei den Adressen vermeiden müssen!

Nun haben also die vier Netzwerkkarten des Routers so wie alle anderen Komponenten unseres Netzes jeweils eine eindeutige IP-Adresse. Die Adresse der Karte, an die ein Teilnetz angeschlossen ist, müssen wir aber noch allen Komponenten des Teilnetzes mitteilen. Sehen wir uns das konkret für Rechner 0.10 an:

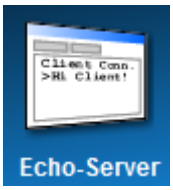


Der entscheidende Punkt ist, dass wir beim Eintrag Gateway die IP-Adresse der Netzwerkkarte des Routers eingetragen haben. Dadurch teilen wir dem Rechner mit, dass er auf diesem Weg sein Netzwerk verlassen

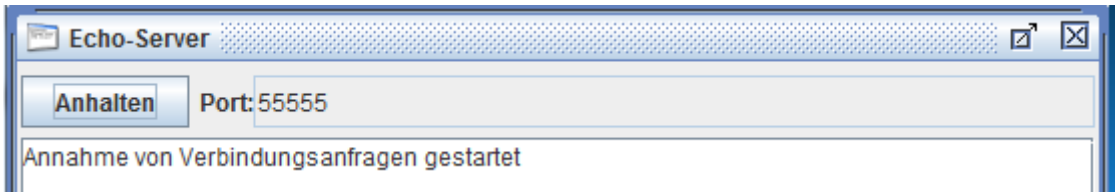
kann. Entsprechend muss bei allen anderen Endgeräten ebenfalls das Gateway konfiguriert werden. Dabei ist zu beachten, dass das Gateway des rechten Teilnetzes eine andere IP-Adresse hat!

Aufgabe 10. *Nimm die oben beschriebenen Konfigurationen am Router und allen Rechnern (inklusive Server) vor.*
Überprüfe erneut mithilfe einer `ping`*-Anweisung die Verbindung von Rechner* 0.10 *zu Rechner* 1.10.
Hinweis: Die Verbindung sollte nun funktionieren. Liegt ein Fehler vor, prüfe noch einmal ganz geduldig alle Einträge in den Konfigurationen. Erfahrungsgemäß macht man sehr schnell irgendwo einen Tippfehler bei der Eingabe der IP-Adressen oder übersieht einen noch offenen Eintrag.

Unser Server erhält nun zum ersten Mal eine echte Server-Aufgabe. Dazu installieren wir auf ihm die Software *Echo Server*:



Anschließend öffnen wir diese Software und klicken auf *Starten*:



Nun installieren wir auf Rechner 1.10 die Software *Einfacher Client* und rufen diese ebenfalls auf. Es sollten nun zwei Fenster geöffnet sein — das des Servers und das des Clients. Jetzt geben wir beim Client die IP-Adresse des Servers an und klicken auf *Verbinden*. Wenn alles richtig gemacht wurde, ist der Client nun mit dem Server verbunden und wir können testweise Nachrichten an den Server schicken. Er sollte sie dann als Echo an uns zurückschicken.

Dies ist neben der `ping`-Anweisung eine weitere Möglichkeit, ein Netzwerk auf Fehler zu überprüfen.

Aufgabe 11. *Falls nicht ohnehin schon geschehen, teste das oben beschrieben Vorgehen selbst mit FILIUS!*

4.2 Die Client-Server-Struktur

Die Unterscheidung zwischen *Client* und *Server* hat sich bei der Kommunikation zwischen Rechnern als sinnvoll erwiesen, so dass wir uns dieser näher anschauen sollten.

Davon, was man unter einem Server versteht, hat wahrscheinlich jeder eine vage Vorstellung, da dieser Begriff auch in der Alltagssprache mittlerweile recht gebräuchlich ist. Der Begriff Client ist einem bisher möglicherweise im Alltag noch nicht begegnet. Kurz gesagt, kann man diese Begriffe so beschreiben:

Server: Ein Server stellt Daten oder Ressourcen zur Verfügung.

Client: Ein Client kann Dienste bei einem Server anfordern.

Umgangssprachlich meint man mit einem Server oft einen Rechner, der entsprechende Aufgaben übernimmt. Streng genommen ist ein Server aber eigentlich nur ein Prozess oder ein Programm! Ein Rechner kann somit, wenn er ein entsprechendes Programm ausführt, die Rolle eines Servers übernehmen. Er kann aber auch jederzeit in die Rolle eines Clients wechseln oder sogar beide Rollen zugleich einnehmen[6]. Um aber ein besseres Verständnis für beide Rollen zu bekommen, versuchen wir solche Mischungen zu vermeiden.

In FILIUS stehen uns die folgenden Server- bzw. Client-Anwendungen zur Verfügung:

Server:

Web-Server: Bereitstellen von Websites.

E-Mail-Server: Verwalten von E-Mail-Konten, Senden, Empfangen, Speichern von E-Mails.

Echo-Server: Spiegeln aller eingehenden Nachrichten an den jeweiligen Client.

DNS-Server: Entschlüsselung symbolischer URLs in IP-Adressen. (Lernen wir noch kennen!)

Client:

Webbrowser: Betrachten von Webseiten.

E-Mail-Programm: Kommunikation mit E-Mail-Server, Lesen und Verfassen von E-Mails.

Einfacher Client: Kontaktaufnahme zu einem Echo-Server.

Mischform:

Gnutella: Peer-To-Peer Anwendung zur Verteilung von Daten in einem Netz[7].

Aufgabe 12. Es gibt noch weitere Typen von Server-Anwendungen, die in FILIUS nicht betrachtet werden. Dazu zählen zum Beispiel:

- Datenbank-Server
- File-Server
- Print-Server
- Proxy-Server

Recherchiere, welche Funktionen diese Server-Anwendungen haben.
Beurteile, ob wir Deiner Erfahrung nach diese in unserem realen Schulnetznetzwerk verwenden.

4.3 Simulation eines Mini-Word Wide Web

4.3.1 Internet vs WWW

Als ersten Schritt zur Simulation des Internets wollen wir eine Miniatur des World Wide Web konstruieren. An dieser Stelle bietet es sich aber vielleicht an, zu klären, was eigentlich der Unterschied zwischen dem *Internet* und dem *World Wide Web* ist. Den meisten Anwendern dürfte nicht bewusst sein, dass es da überhaupt einen Unterschied gibt, zumal in der Alltagssprache zwischen beiden Begriffen kaum unterschieden wird.

Das Internet ist ein weltweiter Verbund von Rechnernetzwerken. Es geht aus dem im Jahr 1969 entstandenen Arpanet hervor, das Forschungszwecken dienen sollte. Über das Internet können eine Vielzahl von *Internetdiensten* in Anspruch genommen werden. Das **World Wide Web (WWW)** ist einer dieser Dienste. Er ist verantwortlich für das Übertragen von Webseiten und damit wohl der Dienst, der von uns am meisten bewusst wahrgenommen wird. Das WWW hat das Internet erst zu dem gemacht, was es heute ist. Zuvor war die Verwendung des WWW eine sehr trockene Angelegenheit — man stelle sich vor, über Kommandozeile im Netz zu surfen.

Wesentlich früher entstand ein anderer berühmter Dienst: Die **E-Mail**. Heute werden diese beide Dienste oft vermischt, da man über Webseiten auf seine E-Mails zugreifen kann. Wer aber zu Hause beispielsweise Thunderbird oder Outlook verwendet, sieht noch ganz klar die Trennung zwischen WWW und E-Mail. Auch im Smartphone sieht man, dass beide Dienste durch unterschiedliche Apps übernommen werden.

Weitere wichtige Internetdienste sind zum Beispiel:

[6] Das ist zum Beispiel bei einer direkten Verbindung zweier Rechner der Fall.

[7] Illegale Tauschbörsen basieren oft auf diesem Prinzip. *Napster* war hier um das Jahr 2000 das erste populäre Beispiel.

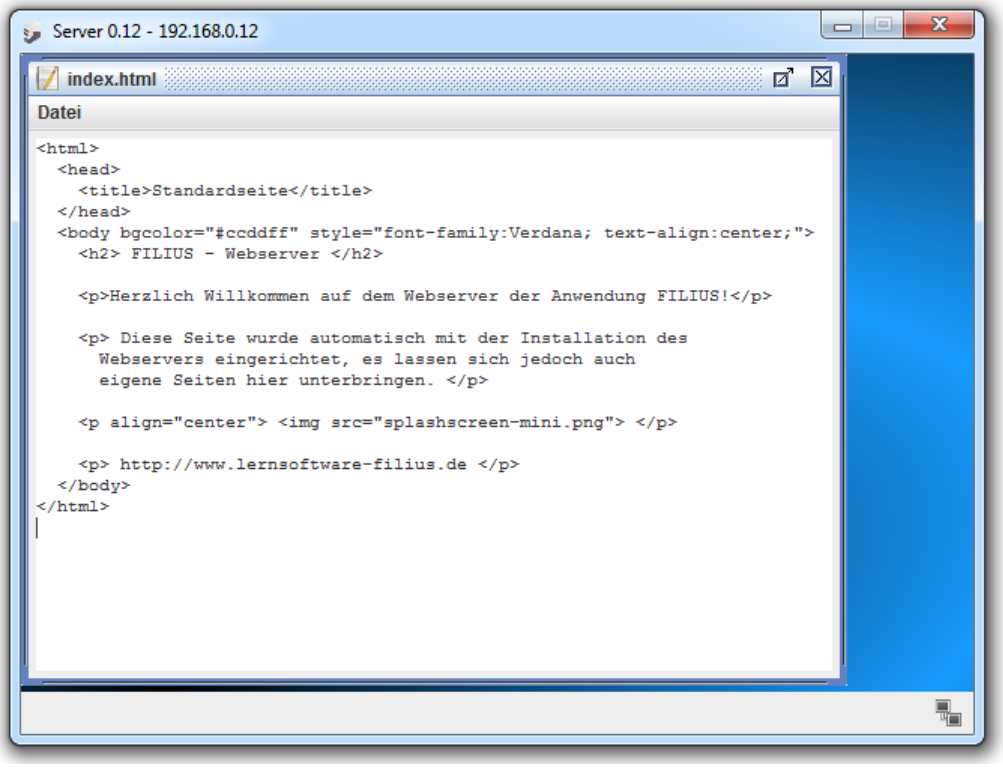
- Dateiverwaltung
- Diskussionsforen
- Chat
- Internettelefonie
- Fernsehen / Livestreams
- Internetradio
- Spiele
- Peer-To-Peer-Systeme

Bei vielen der Dienste ist die Tendenz zu erkennen, dass sie (wie E-Mail) mehr und mehr über das WWW angeboten werden und so die Grenzen verwischen, da man fast alles über den Browser erledigen kann. Letztlich stellt aber das WWW dann lediglich einen Zugang zu einem Dienst dar und ersetzt damit keineswegs den entsprechenden Dienst.

4.3.2 Einrichten eines Webservers

Wir knüpfen hier an unser Netzwerk aus Aufgabe 11 an. Zunächst installieren wir auf dem Server 0.12 die beiden Anwendungen *Webserver* und *Text-Editor*.

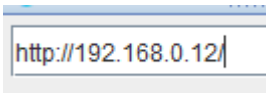
Starten wir den Text-Editor, können wir im Verzeichnis `webserver` die Datei `index.html` finden und öffnen:



Aufgabe 13. *Modifiziere die Datei so, dass Sie einen persönlicheren Begrüßungstext enthält und nicht den Standardtext.*

Nun müssen wir die Anwendung Webserver noch starten. Dazu schließen wir den Text-Editor, öffnen die Anwendung Webserver und klicken *Starten*. Damit steht unsere Seite bereit, um von anderen Rechnern aufgerufen zu werden.

Auf Rechner 1.10 installieren wir jetzt die Anwendung *Webbrowser*. Anschließend starten wir den Browser und geben als Adresse die IP-Adresse des Servers ein:

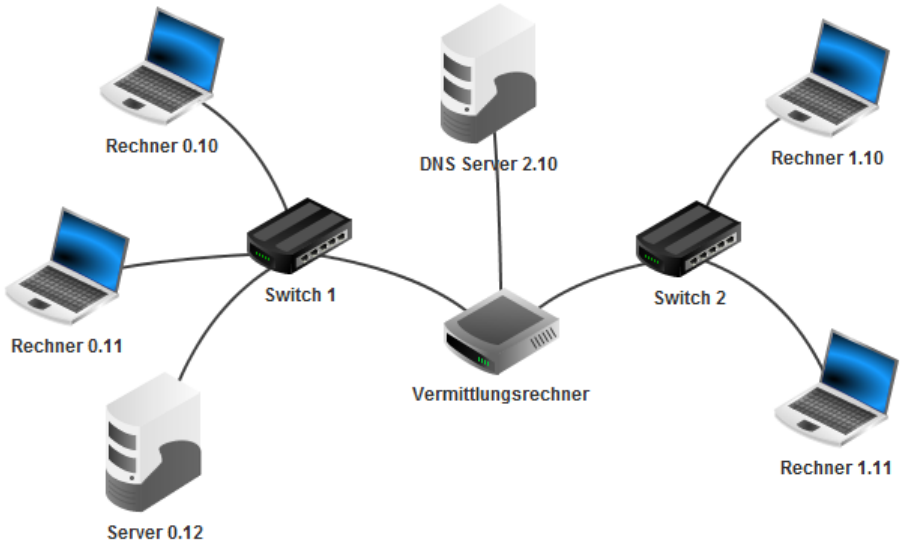


Aufgabe 14. *Rufe auf diesem Weg die Webseite im Browser auf und beobachte dabei auch, was der Webserver anzeigt.*

Die Anzeige des Webservers soll einen ersten Eindruck davon vermitteln, was im Hintergrund beim einfachen Aufrufen einer Seite alles abläuft. Genauer werden wir uns dies noch später ansehen.

4.3.3 Domain Name Server

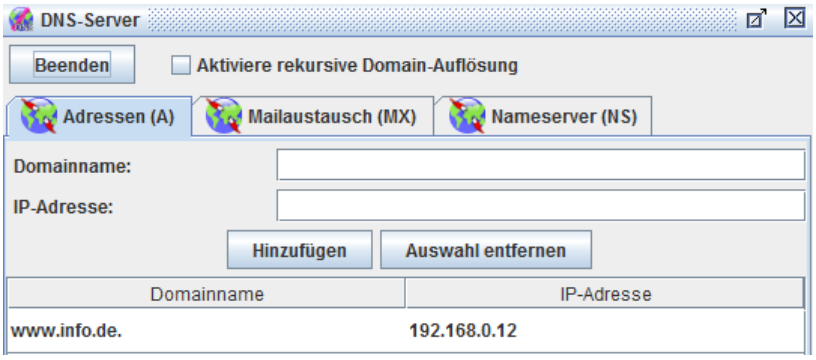
Das Aufrufen einer Webseite mittels IP-Adresse ist natürlich recht ungewohnt und wenig komfortabel. Daher wollen wir nun dafür sorgen, dass unsere Seite unter der Adresse `www.info.de` gefunden werden kann. Um dies zu erreichen, benötigen wir einen *Domain Name Server (DNS).*



Aufgabe 15. *Verbinde mit dem Router einen weiteren Server. Dieser soll die IP-Adresse* 192.168.2.10 *erhalten. Achte beim Verbinden darauf, dass Du beim neuen Server das Gateway einstellen musst!*
Teste mittels `ping`*, ob der neue Server erreichbar ist.*

Ist der Server erfolgreich hinzugefügt worden, können wir bei **allen** weiteren weiteren Rechnern in der Konfiguration unter Domain Name Server die IP-Adresse des neuen Servers angeben.

Schließlich müssen wir beim neuen Server die Anwendung *DNS-Server* installieren und öffnen. Wir tragen dort den gewünschten Domainnamen `www.info.de` und die entsprechende IP-Adresse ein, klicken auf *Hinzufügen* und starten den Dienst:

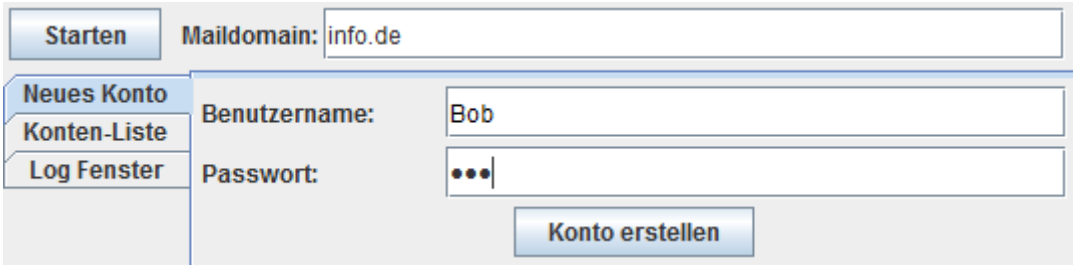


Aufgabe 16. *Starte in Rechner* 1.10 *den Webbrowser und teste, ob unsere Seite tatsächlich unter* `www.info.de` *gefunden wird. Falls nicht, musst Du leider noch einmal alle Konfigurationen genau prüfen.*

Im realen Netz ist die Situation natürlich etwas komplizierter, da nicht alle Webadressen auf einem zentralen Server gespeichert sind. Aber die Grundidee, dass Webadressen mithilfe solcher Server *aufgelöst* werden, haben nur nun schon einmal kennengelernt.

4.4 E-Mail

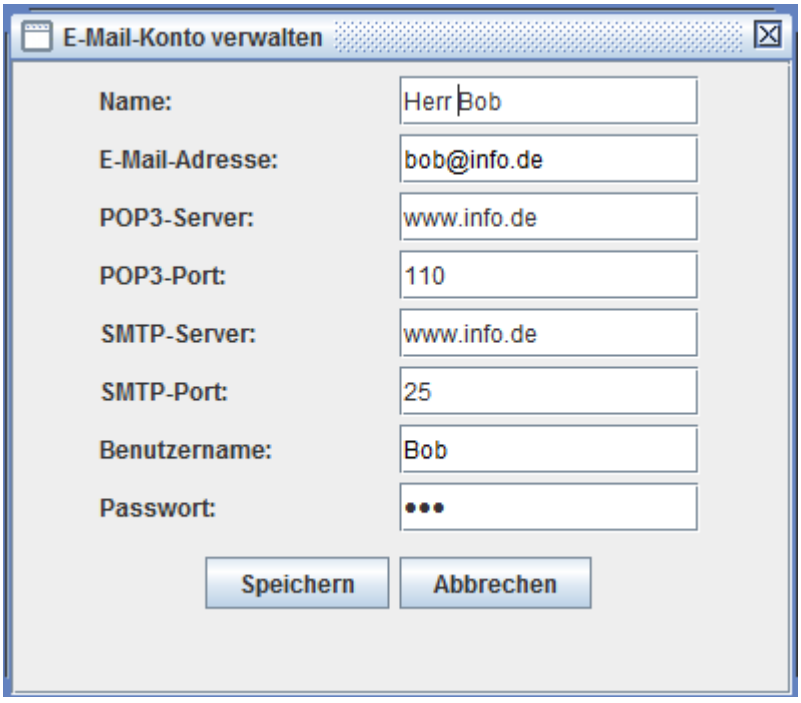
Als weiteren wichtigen Internetdienst wollen wir nun den E-Mail-Verkehr simulieren. Dazu installieren wir auf dem Server 0.12 die Anwendung *Email-Server* und öffnen diese sogleich. Nun können wir die Maildomain festlegen und ein neues Konto anlegen. Als Domain verwenden wir `info.de`, als Benutzernamen `Bob` und als Passwort `bob`:



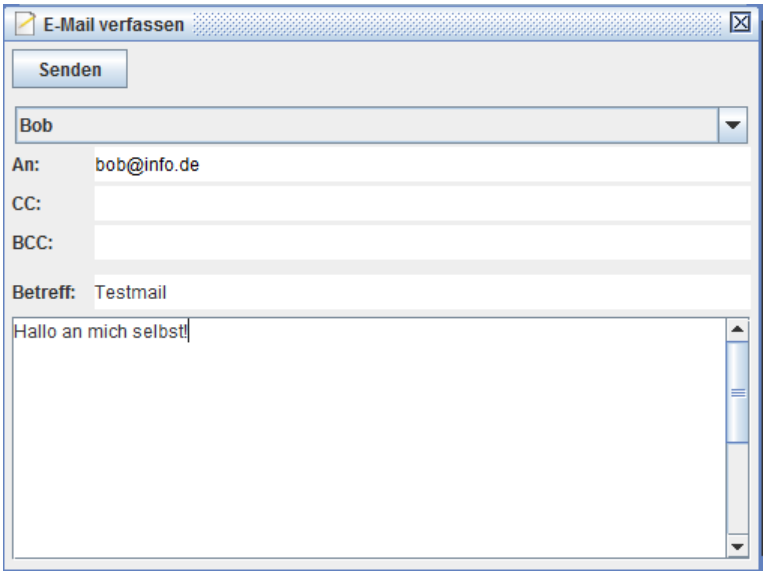
Danach starten wir den Dienst.

Jetzt muss die eben gewählte Maildomain beim DNS-Server eingerichtet werden. Wir öffnen also die Anwendung *DNS-Server* auf Server 2.10 und fügen dort unter *Mailaustausch* die entsprechenden Einträge hinzu:

Schließlich müssen wir noch das *E-Mail-Programm* auf einem Rechner installieren. Zum Test installieren wir es auf Rechner 0.10 und öffnen es. Wir klicken auf *Konto einrichten* und geben Bobs Zugangsdaten ein.



Die Nummern bei den Ports — ihre Bedeutung erläutern wir gleich — sind übrigens Standardwerte. Wir legen das Konto an und schreiben zur Probe sogleich eine E-Mail:



Aufgabe 17. *Teste, ob das Schreiben einer E-Mail funktioniert! Wenn nicht, heißt es leider wieder, dass alle Einstellungen nochmal genau geprüft werden müssen.*

Aufgabe 18. *Richte auf dem E-Mail-Server eine weitere E-Mail-Adresse* `bert@info.de` *ein.*
Installiere auf Rechner 0.11 *das E-Mail-Programm für Bert.*
Danach teste, ob sich die beiden gegenseitig E-Mails schreiben können. Teste dabei beide Richtungen!

In der nächsten Aufgabe werden absichtlich nicht alle Zwischenschritte erwähnt. Alle nötigen Schritte sind wir aber eben bereits einmal durchlaufen!

Aufgabe 19. *Ergänze nun im rechten Netzwerk einen E-Mail-Server* 1.13 *mit der Maildomain* `mail.de`.
Füge auf diesem ein Konto `alice@mail.de` *hinzu.*
Richte für Alice auf Rechner 1.10 *das E-Mail-Programm ein und teste den Versand zwischen Alice und Bob.*

4.5 Ports

Wir wissen bereits, dass in einem Netzwerk jedes Gerät eindeutig durch seine IP-Adresse identifizierbar ist. Analog dazu, ist eine Firma, der wir einen Brief schreiben möchten durch ihren Namen und ihre Anschrift eindeutig identifizierbar. Oft muss (oder sollte man zumindest) noch eine genauere Angabe machen — zum Beispiel, an welche Abteilung innerhalb der Firma der Brief weitergeleitet werden soll oder an welchen ganz bestimmten Mitarbeiter.

Diese feinere Unterscheidung spiegelt sich in Netzwerken in der *Portnummer* wider. Statt verschiedener Abteilungen oder Mitarbeiter gibt es hier verschiedene Anwendungen oder Prozesse, die Daten empfangen sollen. Bei einer Verbindung zweier Anwendungen auf verschiedenen Rechnern können sich die beiden auf einen Port einigen, damit die Daten gezielt an richtige Anwendung weitergeleitet werden können.

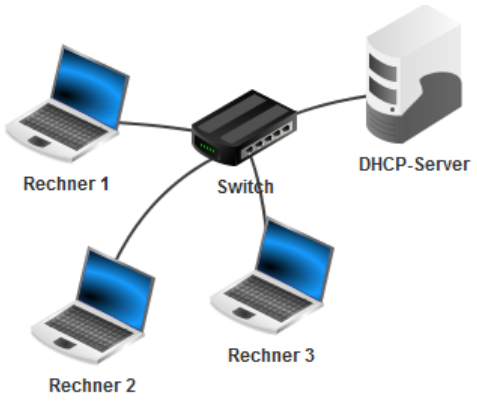
Es gibt aber auch einige Standardwerte, sogenannte *well-known ports*, die für bestimmte Zwecke reserviert sind. Port 25 ist beispielsweise für E-Mail-Verkehr reserviert oder Port 80 für den Empfang von Webseiten.

Die meisten Anwendungen erledigen erfreulicherweise die Einstellung der Ports automatisch, so dass man als Endnutzer selten oder nie damit in Berührung kommt.

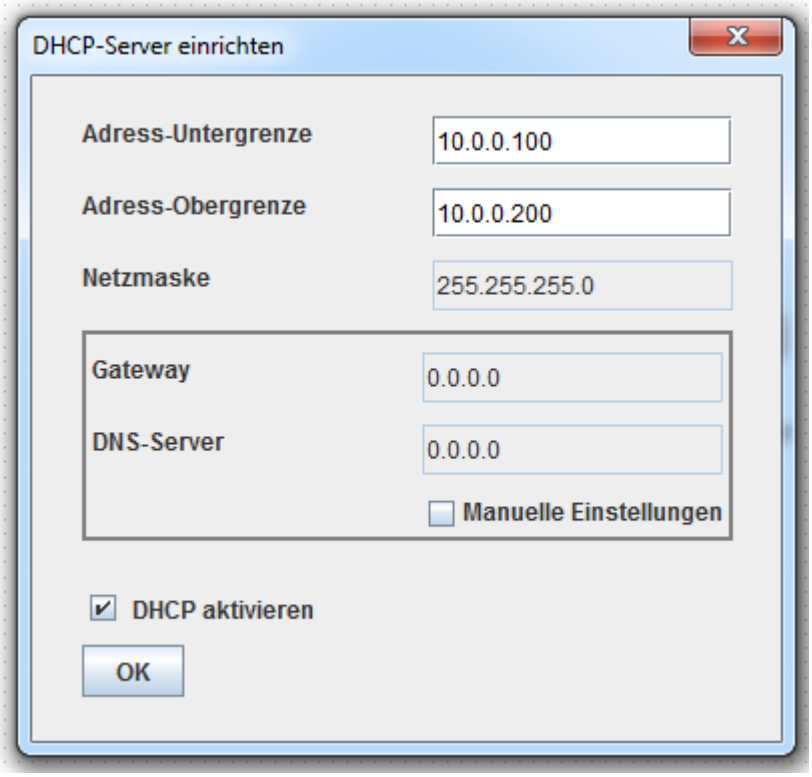
5 Dynamische IP-Adressen

Höchstwahrscheinlich haben die wenigsten von uns an ihrem Rechner oder Telefon jemals eine IP-Adresse festlegen müssen. Glücklicherweise wird dies nämlich in vielen Netzwerken (vor allem zu Hause) automatisch erledigt. Simulieren wir einmal ein kleines Heimnetzwerk mit automatischer Vergabe von IP-Adressen.

Erstellen wir dazu zunächst ein Netzwerk mit drei Rechnern und einem Server wie in der Abbildung:



Dem Server teilen wir die IP-Adresse 10.0.0.10 zu und klicken anschließend auf den Button *DHCP-Server einrichten*. Wir geben Adress-Unter- und Obergrenze ein und machen einen Haken bei *DHCP aktivieren*:



Nun weiß der Server, dass er für die Zuteilung der IP-Adressen zuständig ist und welchen Spielraum er bei der Wahl der Adressen hat. Zu Hause übernimmt übrigens normalerweise der Router die Rolle des DHCP-Servers und nicht etwa einer der Rechner, die man daheim stehen hat.

Den Rechnern müssen wir noch mitteilen, dass sie automatisch eine IP-Adresse beziehen sollen. Dazu setzen bei wir bei allen Rechnern in der Konfiguration einen Haken bei *DHCP zur Konfiguration verwenden.*

Aufgabe 20. *Wechsle in den Aktionsmodus und beobachte die Kommunikation zwischen den Rechnern. Man sollte erkennen, dass alle Rechner den Server kontaktieren, um eine IP zu erhalten.*
Ermittle anschließend, welche IP-Adressen ihnen zugeteilt wurden. (Einen möglichen Weg, um sie herauszufinden, haben wir in einem der ersten Abschnitte kennengelernt!)

Aufgabe 21. Neben dem Komfort, nicht manuel IP-Adressen eingeben zu müssen, gibt es noch einen weiteren Vorteil von dynamischen IP-Adressen.
Erläutere, was für ein weiterer Vorteil denkbar ist und in welchen Netzwerken dieser besonders zum Tragen kommt.

6 Netzprotokolle

6.1 Grundideen von Schichtenmodellen und Protokollen

Beginnen wir zunächst mit einem (natürlich etwas gestellten) anschaulichen Beispiel, um die Idee eines Schichtenmodells einzuführen.

Stellen wir uns dazu vor, wir sind in der längst vergangenen Zeit, in der zum ersten Mal ein japanisches Unternehmen in Düsseldorf einen Sitz eröffnete. Zu der Zeit gab es noch keine Kommunikation über das Internet, stattdessen musste die Kommunikation zwischen den Kontinenten noch ganz altmodisch per Morsecode stattfinden.

Nun möchte der Chef des düsseldorfer Standortes eine Nachricht an die Konzernzentrale in Japan schicken. Dieser Vorgang könnte wie folgt ablaufen:

(1) Der Chef diktiert seine Nachricht.

(2) Seine Sekretärin schreibt die Nachricht mit.

(3) Ein Übersetzer übersetzt die Nachricht von Deutsch auf Englisch (weil vor Ort niemand Japanisch kann).

(4) Ein Funker sendet per Morsecode die englische Nachricht nach Japan.

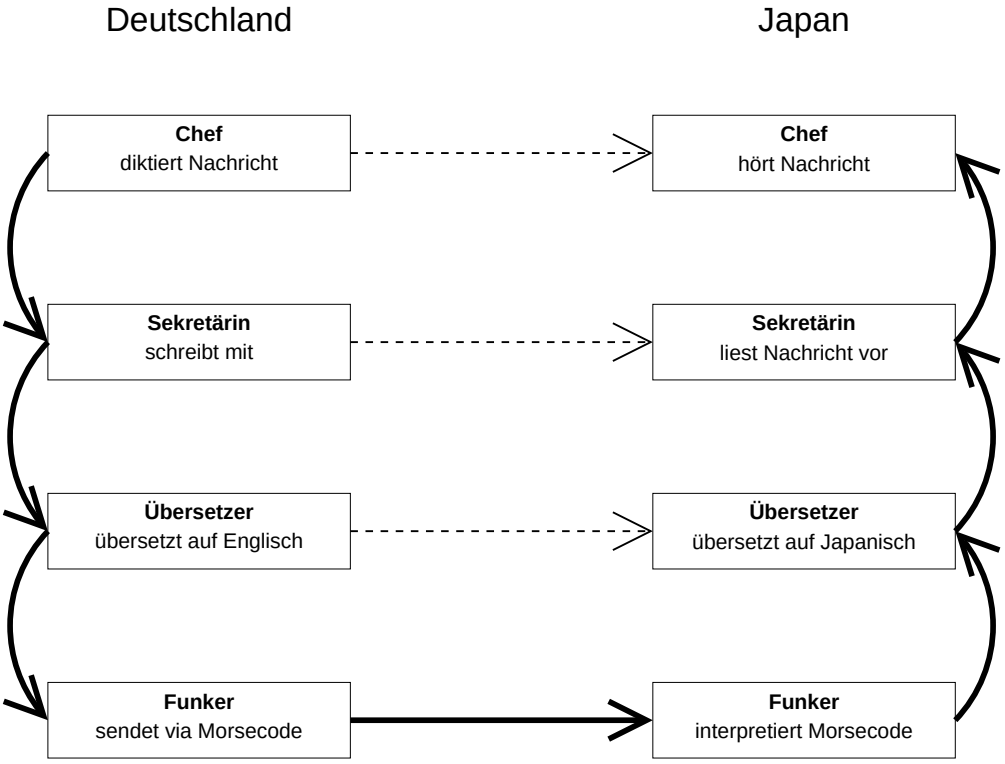
(5) Der Funker in Japan empfängt den Morsecode und schreibt die Nachricht mit.

(6) Ein Übersetzer übersetzt die Nachricht von Englisch auf Japanisch.

(7) Die Sekretärin des Chefs liest ihm die Nachricht vor.

(8) Der Chef hört sich die Nachricht an.

Sehen wir uns diesen Vorgang einmal in einem Diagramm an:



Die Nachricht durchläuft hier auf der linken Seite zunächst mehrere *Schichten*. Jede Schicht (bis auf die oberste) stellt der Schicht über ihr eine bestimme Funktionalität bereit. So bietet beispielsweise der Übersetzer die Möglichkeit an, eine Nachricht ins Englische zu übersetzen. Gleichzeitig macht jede Schicht (bis auf die unterste) Gebrauch von den Diensten der darunterliegenden Schicht. Wie eine Schicht ihre Aufgabe im Detail erledigt, ist den anderen gleichgültig.

Auf der rechten Seite wandert die Nachricht zwar von unten nach oben, aber auch dort bietet die jeweils weiter unten liegende Schicht der oberen ihre Funktionalität an. Zum Beispiel bietet der Übersetzer dort dem der Sekretärin die Möglichkeit, eine Nachricht für sie ins Japanische zu übersetzen. Die weiter oben liegenden Schichten machen also auch hier Gebrauch vom Angebot der darunter liegenden.

Die eigentliche Übertragung der Nachricht findet in der untersten Ebene statt. Aus Sicht jeder einzelnen Schicht findet aber (indirekt) ebenfalls eine horizontale Übertragung statt. Der deutsche Chef hat beispielsweise den Eindruck, dass er etwas diktiert, was der japanische Chef dann hört. Die deutsche Sekretärin schreibt etwas auf, was die japanische Sekretärin (vor)liest. Dass die Nachricht zunächst noch in die vertikale Richtung weiter wandert, muss sie nicht interessieren. Man sagt, dass zwischen den beiden Chefs, den beiden Sekretärinnen und den beiden Übersetzern jeweils eine *logische Verbindung* besteht, im Gegensatz zu der realen Verbindung zwischen den beiden Funkern.

Es ist denkbar, dass auf jeder Ebene ein gewisses *Protokoll* zur Kommunikation verwendet wird. Am deutlichsten ist dies vielleicht bei der untersten und obersten Ebene nachvollziehbar. Die Funker müssen sich bei der Verwendung von Morsecode genau an einen bestimmten Ablauf halten und neben dem eigentlichen Text möglicherweise zusätzliche Informationen senden. Vielleicht müssen sie Beginn und Ende der Nachricht markieren oder bei unklarer Übertragung ein erneutes Senden erbeten. Die Chefs haben untereinander vielleicht ein recht genaues Protokoll zum Umgang miteinander. Dieses könnte zum Beispiel festlegen, wie sie sich gegenseitig ansprechen und wie sie eine Nachricht beenden.

Insgesamt sollte hier deutlich werden, dass der gesamte Vorgang recht komplex ist. Die Zerlegung in Schichten soll dazu dienen, ihn überschaubarer zu machen und die einzelnen Schritte unabhängig voneinander mithilfe ihrer Protokolle planen zu können.

Aufgabe 22. Das Zerlegen eines Vorgangs oder eines Sachverhalts in verschiedene Schichten oder Abstraktionsebenen ist uns in der Informatik schon an anderer Stelle begegnet.
Nenne ein Beispiel für ein weiteres Schichtenmodell in der (Schul-)Informatik.

Aufgabe 23. Vielleicht ohne uns darüber bewusst zu sein, verwenden wir auch im Alltag häufig Protokolle für unsere Kommunikation. Ein klassisches Beispiel ist das ganz normale Telefonat mit einem Festnetztelefon. Für gewöhnlich wird es damit eröffnet, dass der Angerufene sich mit seinem Namen meldet.

(a) *Beschreibe den weiteren Ablauf eines Telefonats.*
Anregungen: Was geschieht nachdem sich der Angerufene gemeldet hat? Auf welche Details sollte man während eines Gesprächs achten? Was kann während eines Telefonats zu Missverständnissen führen? Woher weiß man, ob der andere zuhört? Stelle Dir am besten vor, Du musst den Ablauf einem Zeitreisenden aus der Vergangenheit erklären, der noch nie ein Telefon gesehen hat!

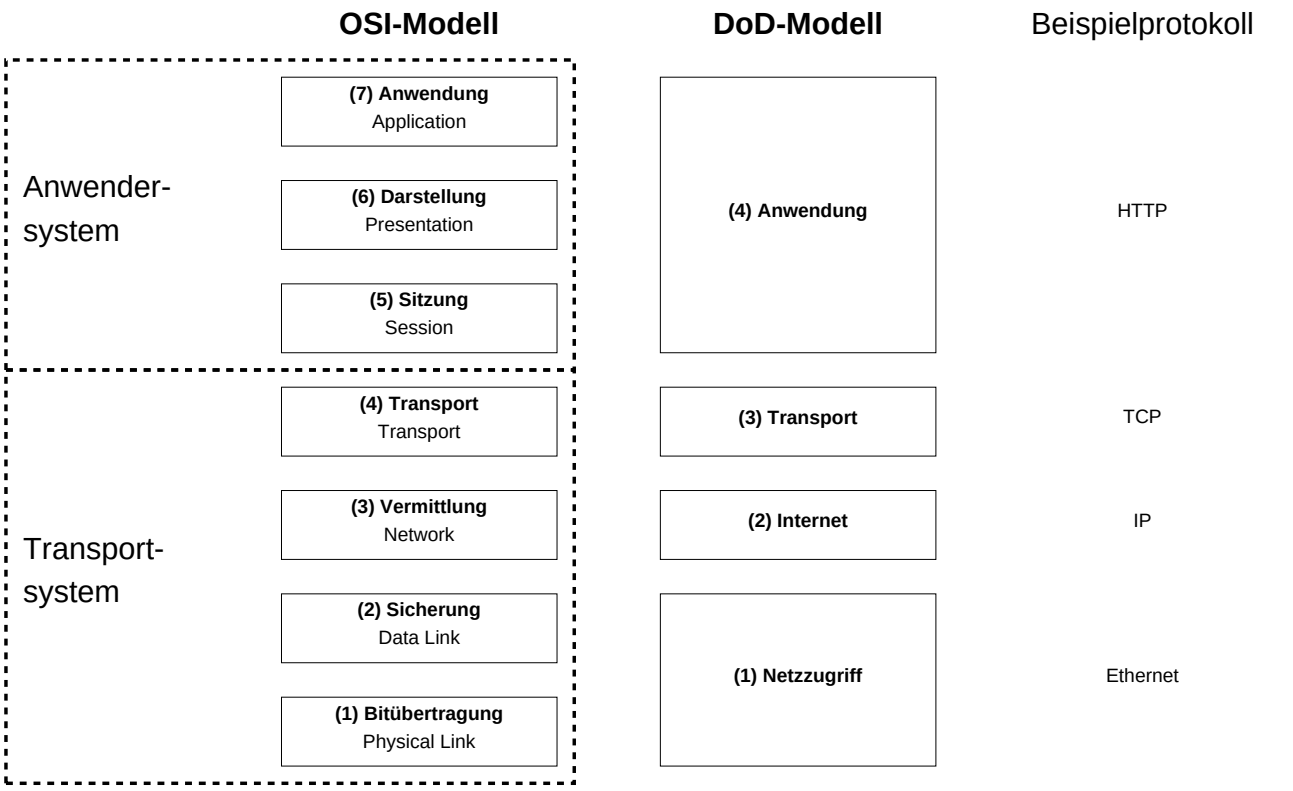
(b) *Nenne wenigstens zwei weitere Situationen, in denen zur Kommunikation ein Protokoll verwendet wird.*
Anregungen: Es können zum Beispiel Situationen aus der Freizeit, der Schule, der Arbeitswelt oder auch dem Fernsehen sein.

(c) *Beschreibe möglichst genau die Protokolle in den von Dir genannten Situationen.*

6.2 OSI- und DoD-Referenzmodelle

Die Kommunikation über ein Datennetz oder insbesondere das Internet ist ein noch wesentlich komplexerer Vorgang als der oben als Einstieg beschriebene. Um die dabei zu bewältigen Problemstellungen lösen zu können, hat es sich als sehr hilfreich erwiesen, diesen Vorgang ebenfalls in verschiedene Schichten zu zerlegen, um die Schwierigkeiten in den einzelnen Schichten getrennt und unabhängig voneinander lösen zu können.

Die Aufteilung in verschiedene Schichten ist nicht immer einheitlich. Es gibt hier einerseits das OSI-Modell (*Open Systems Interconnection Model*), das 1984 von der International Organization for Standardization (ISO) als Standard veröffentlicht wurde. Das wichtigste alternative Modell ist das bereits in der 1960er Jahren entstandene DoD-Modell (*Department of Defense Intenet Architecture Model*). Dieses ist, wie man am Diagramm erkennen kann, in weniger Schichten unterteilt und daher für uns vielleicht etwas leichter nachzuvollziehen:



Welches der beiden Modelle nun das bessere ist, ist — wie oft in der Informatik — ein gern diskutiertes Thema. Für uns soll diese Frage zweitrangig sein. Viel wichtiger ist, dass wir das grundlegende Konzept der Verwendung eines Schichtenmodells verstehen.

Im Diagramm sind für die Schichten des DoD-Modells bereits beispielhaft vier Protokolle angegeben. Mit diesen können Webseiten übertragen werden. Im OSI-Modell könnte man diese Situation noch feiner unterteilen. Beispielsweise könnte in der Anwendungsschicht ein Web Browser genannt werden, in der Darstellungsschicht der HTML-Standard und in der Sitzungsschicht letztlich das HTTP-Protokoll.

6.3 Die TCP/IP-Protokolle

Die beiden Protokolle TCP und IP bilden zusammen das Übertragungsprotokoll des Internets. Wir haben in den beiden Schichtenmodellen schon gesehen, dass sie unter der Sitzungs- bzw. Anwendungsschicht angesiedelt sind. Sie können also beispielsweise von einer Anwendung wie einem E-Mail-Client eine Datei übergeben bekommen, die sie dann versenden müssen.

Um die Arbeitsweise besser nachvollziehen zu können, stellen wir uns zunächst folgende Situation vor. Wir haben bei einem Möbelhaus eine neue Wohnungseinrichtung bestellt und darum gebeten, dass diese geliefert werden soll. Natürlich können wir nicht erwarten, dass die gesamte Einrichtung in einem einzigen großen Karton bei uns vor der Türe landet — wer sollte den schon tragen können? Stattdessen wäre folgender Ablauf denkbar:

(1) Ein Mitarbeiter teilt zunächst die Bestellung in kleinere Kartons von tragbarer Größe auf und beschriftet diese so, dass erkennbar ist, in welchem Zimmer sie später abgestellt werden sollen und aus welcher Abteilung des Möbelhauses der Inhalt stammt. Es wäre schließlich sehr mühselig, wenn wir erst in jeden Karton hineinsehen müssten, um dann festzustellen, wo er hingehört. Außerdem beschriftet er sie mit fortlaufenden Nummern. Danach übergibt er die Kartons an die Versandzentrale.

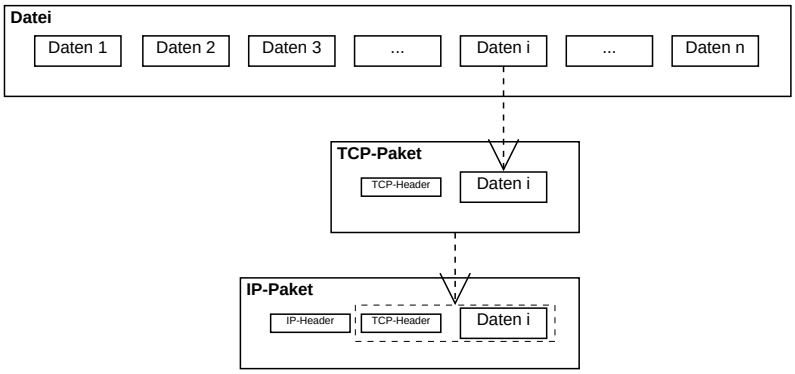
(2) In der Versandzentrale beschriftet ein weiterer Mitarbeiter die Pakete mit seiner und unserer Anschrift, also mit Absender- und Empfängeradresse. Danach holt der Paketdienst die Lieferung ab.

Die verschiedenen Pakete müssen nicht alle auf demselben Weg zu uns gelangen. Vielleicht landen sie in verschiedenen LKW, die unterschiedliche Routen einschlagen. Das hat den Vorteil, dass im Falle eines Staus oder einer Straßensperrung zumindest ein Teil der Lieferung zeitnah ankommt. Fehlt in unserer Lieferung etwas, können wir dies leicht anhand der fortlaufenden Nummern erkennen und ggf. eine erneute Sendung anfordern, falls auch nach längerer Wartezeit die Lieferung nicht vollständig ist.

In der oben dargestellten Situation hat der erste Mitarbeiter die Rolle des TCP-Protokolls, der zweite die des IP-Protokolls. Statt einer Wohnungseinrichtung sollen diese Protokolle natürlich eine Datei verschicken.

Zunächst erhält also das TCP-Protokoll die Datei von der darüber liegenden Schicht. Wie im Gedankenspiel wird auch hier die Lieferung, also die Datei, zerstückelt und in sogenannte *Datenpakete* zerlegt. So ein Paket

kann beispielsweise 4 kB groß sein. Die Beschriftung dieser Pakete wird hier bewerkstelligt, indem jedem Paket ein *TCP-Header* hinzugefügt wird:



Dieser Header enthält unter Anderem die folgenden Informationen:

- Portnummer des empfangenden Programms (kurz *DP* für *Destination Port*)
- Portnummer des sendenden Programms (kurz *SA* für *Source Port*)
- Anzahl der Bytes, die in die Richtung des aktuellen Pakets schon gesendet wurden (kurz *SN* für *Sequence Number*)
- Anzahl der Bytes, die in der umgekehrten Richtung angekommen sind (kurz *AN* für *Acknowledge Number*)

Die letzten beiden genannten Einträge dienen dem Aufspüren von Übertragungsfehlern. Außerdem enthält der Header noch eine Prüfsumme, die ebenfalls dabei helfen soll, fehlerhafte Übertragungen zu bemerken. Beachte, dass all diese Angaben für den empfangenden Rechner interessant sind. Für den eigentlichen Versandweg sind sie nicht relevant.

Die fertigen TCP-Pakete werden dann an das IP-Protokoll übergeben, das im Gedankenspiel von oben den zweiten Mitarbeiter darstellt. Dieses Protokoll ist für den Versand einzelner Pakete verantwortlich. Es kümmert sich also nicht darum, welche Pakete zusammengehören, sondern versucht jedes einzelne so effizient wie möglich seinem Empfänger zukommen zu lassen.

Auch das IP-Protokoll fügt jedem Paket wieder einen Header hinzu, wie man im Diagramm oben bereits erkennen kann. Der IP-Header enthält unter Anderem die folgende Angaben:

- Zieladresse (kurz *DA* für *Destination Address*)
- Absenderadresse (kurz *SA* für *Source Address*)
- Lebenszeit (kurz *TTL* für *Time to live*)

Die Lebenszeit ist eine Zahl, die jedes Mal, wenn das Paket von einem Vermittlungsrechner weitergeleitet wird, um 1 verringert wird. Erreicht sie den Wert 0, wird das Paket gelöscht. Dadurch sollen Irrläufer aufgefangen werden. Es sollte hier auffallen, dass diese Angaben im Gegensatz zu den Angaben im TCP-Header für die am Versand beteiligten Systeme (Vermittlungsrechner, Router,…) wichtig sind.

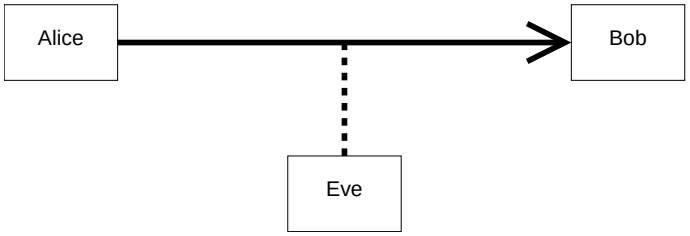
Hat der Empfänger-Rechner ein Paket entgegengenommen, ist die Aufgabe des IP-Protokolls für dieses Paket erledigt. Das Paket wird dann an TCP hoch gereicht. Auf der Empfängerseite ist das TCP-Protokoll nun auch dafür verantwortlich, die Pakete wieder in die richtige Reihenfolge zu bringen oder möglicherweise fehlende Pakete erneut anzufordern.

7 Verschlüsselungsverfahren

7.1 Einführung

Das Grundproblem der Kryptographie ist schnell erklärt und wird sicher jedem einleuchten. Zwei Parteien **A** und **B** möchten miteinander auf einem Kanal kommunizieren, bei dem sie sich nicht sicher sein können, ob jemand ihrer Kommunikation lauscht. Sie möchten aber keinesfalls, dass dieser Lauscher ihre Kommunikation versteht.

Statt **A** und **B** haben sich in der Kryptographie die Namen Alice und Bob eingebürgert. Der Lauscher heißt meist Eve (für *eavesdropper*):



Dass Kryptographie im Zeitalter des Internets ein hochaktuelles Thema ist, wird niemand bezweifeln. Während die meisten Privatanwender mit dem Thema — ob zu Recht oder Unrecht sei dahingestellt — aber eher locker umgehen, kann es für Firmen hohe Verluste bedeuten, wenn ein Konkurrent wertvolle firmeninterne Informationen erhält. Von noch viel höherer Wichtigkeit ist ein zuverlässiges Kryptosystem im militärischen Bereich. Dieser hat schon lange bevor es Rechner oder gar das Internet gab maßgeblich zur Entwicklung der Kryptographie beigetragen.

Grundsätzlich unterscheidet man zwei Arten von Kryptosystemen: Symmetrische und asymmetrische. Bei *symmetrischen* Verfahren benötigt man zum Ver- und Entschlüsseln denselben Schlüssel. Das bedeutet letztlich, dass Alice und Bob einen gemeinsamen Schlüssel brauchen, um zu kommunizieren. Bei *asymmetrischen* Verfahren hingegen benötigt man zum Entschlüsseln einen anderen Schlüssel als zum Verschlüsseln. Man kann bei diesen Verfahren den Schlüssel zum Entschlüsseln auch nicht aus dem zum Verschlüsseln berechnen. D.h. Alice hätte einen Schlüssel zum Verschlüsseln und Bob den dazu passenden zum Entschlüsseln. Alice kann also Nachrichten nur verschlüsseln und selbst die verschlüsselten Nachrichten nicht wieder entschlüsseln!

Wie so ein asymmetrisches Verfahren aussehen könnte, ist im Moment vielleicht nicht so recht klar. In der Tat war die Entwicklung solcher Systeme ein historisch sehr bedeutender Schritt in der Kryptographie. Diese Systeme verwenden einige mathematische Tricks und Sachverhalte, die einem aus der Schule nicht bekannt sind. Wir werden aber dennoch eines dieser Verfahren betrachten und so weit es geht nachvollziehen.

Aufgabe 24. *Beschreibe Situationen, in denen Kryptographie Deiner Meinung nach verwendet wird oder verwendet werden sollte. Nenne dabei auch Bereiche, die Dich unmittelbar betreffen.*

7.2 Symmetrische Verfahren

7.2.1 Caesar

Das Caesar-Verfahren ist das wohl am einfachsten nachvollziehbare Verschlüsselungsverfahren. Es ist benannt nach Gaius Julius Caesar, der dieses Verfahren für seine militärische Korrespondenz verwendet hat. Einfach ausgedrückt, besteht es aus einer Verschiebung des Alphabets. In der folgenden Tabelle wird als Beispiel eine Verschiebung um 4 Zeichen dargestellt:

Plain Text	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Cipher	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D

Das Klartextwort[8] „HALLO“ würde demnach verschlüsselt zu „LEPPS“. Anstatt zu sagen, dass hier eine Verschiebung von 4 Zeichen stattgefunden hat, könnte man übrigens auch sagen, dass der Schlüssel **E** verwendet wurde.

Aufgabe 25. Gegeben ist der Klartext ANGRIFF BEI SONNENAUFGANG.

(a) *Verschlüssle diesen Text mit dem Schlüssel **E**.*

(b) *Verschlüssle diesen Text mit dem Schlüssel **L**.*

[8] Im Deutschen sagt man *Klarttext* für *Plain Text*, *Schlüssel* für *Key* und *Schlüsseltext* für *Cipher*. Da aber der letzte Ausdruck manchmal etwas verwirrend sein kann — vor allem, wenn der Schlüssel selbst ein Text ist — werden hier neben den deutschen oft auch die englischen Begriffe verwendet.

Aufgabe 26. Gegeben ist der Schlüsseltext GLH VFKODKW LVW JHZRQQHQ.
Analysiere diesen Text und versuche, den Klartext zu ermitteln.
Erläutere mögliche Ansätze, die Dir dabei in den Sinn kommen.

Aufgabe 27. *Beurteile die Sicherheit des Caeser-Verfahrens.*

Man kann das Caesar-Verfahren auch etwas mathematischer beschreiben. Das ist zwar nicht notwendig, bietet uns aber eine gute Vorübung, um später folgende Verschlüsselungsverfahren besser nachvollziehen zu können. Diese werden nämlich um einiges mathematiklastiger sein.

Um ein Verschlüsselungsverfahren mathematisch zu beschreiben, bietet es sich an, statt Buchstaben bzw. Texte lieber Zahlen bzw. Zahlenfolgen zu verschlüsseln. Letztlich ist es im Grunde ohnehin dasselbe, denn man kann mit einer kleinen Tabelle Buchstaben in Zahlen übersetzen:

Buchstabe	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Zahl	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Das Verschlüsseln wird nun durch eine Funktion beschrieben. Nehmen wir als Beispiel die Verschiebung um 4 Zeichen, die wir auch als Einstieg betrachtet haben, und überlegen wir uns, wie diese Funktion aussehen könnte. Vielleicht bekommen wir eine Idee, wenn wir uns die zugehörige Tabelle zur Verschlüsselung ansehen:

Klartext	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Schlüsseltext	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	0	1	2	3

Aufgabe 28. Gesucht ist also nun eine Funktion f_4, die jeder Klartextzahl x die Schlüsseltextzahl $f_4(x)$ zuordnet, z.B. $f_4(3) = 7$. Die kleine Vier steht dabei für die hier verwendete Verschiebung um 4 Stellen.

(a) *Ermittle eine geeignete Funktion f_4.*
Tipp: Du benötigst dazu eine Grundrechenart und eine Rechenoperation, die wir in Informatik vor einiger Zeit kennengelernt haben.

(b) *Ermittle eine geeignete Umkehrfunktion g_4 zum Entschlüsseln.*

Aufgabe 29. *Verallgemeinere Deine Ergebnisse aus der letzten Aufgabe. D.h. ermittle für eine Verschiebung um a Stellen eine Verschlüsselungsfunktion f_a und eine Entschlüsselungsfunktion g_a.*

7.2.2 Substitution

Das Caesar-Verfahren ist ein spezielles Beispiel für sogenannte *Substitutionsverfahren*. Bei diesen Verfahren wird jeder Buchstabe durch einen anderen ersetzt. Während dieses Ersetzen beim Caeser-Verfahren nach einem schnell durchschaubaren Muster stattfindet, darf man bei einem allgemeinen Substitutionsverfahren vollkommen frei wählen, welcher Buchstabe wie ersetzt wird. Zum Beispiel könnte man wie folgt substituieren:

Plain Text	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Cipher	D	F	Z	G	E	I	Y	J	L	H	Q	M	O	K	P	S	U	N	V	X	W	R	T	B	C	A

Man sagt dann auch, dass man das Schlüsselwort DFZGEIYJLHQMOKPSUNVXWRTBCA verwendet.

Aufgabe 30. *Ermittle, wie viele verschiedene Schlüsselwörter zur Verschlüsselung mit einem Substitutionsverfahren zur Verfügung stehen.*
Hinweis: Die Zahl ist sehr(!) groß.

Angesichts der enorm großen Zahl an denkbaren Schlüsselwörtern könnte man annehmen, dass Eve wenig Chancen hat, einen mit einem Substitutionsverfahren verschlüsselten Text zu entschlüsseln. Dem ist allerdings nicht so. Beispielsweise kann schon ein einfaches Werkzeug wie dieses

`http://www.kas-bc.de/krypto/analyse.php`

bei der Entschlüsselung sehr hilfreich sein. Wie genau es helfen kann, sollt Ihr Euch allerdings selbst überlegen.

Aufgabe 31. Gegeben ist die folgende verschlüsselte Nachricht:

GEKQEKWKGXWKXWKWKGGEKQEKGDVLVXGLEVWOOEDMMENTELVJELXRPKHEJENDKENQ
DKKXRPKHEJENYEFXKLZJXELKYEVEJEKRPKELKEOHEGEKFELGEVOWVVTLEDWVWKGE
LKDXOEKVLZJLOMEFEKETLYIPNXJLKWKGTLGENFETEYEKTLEINDYEWKGDKXTPNXVP
MMXEELKVPJKEGDVDKGNEKLZJXVXDXXILKGEKTENVLZJAWOYEVEXAODZJXTDVELKE
OHEGEKKEWYEFPNKEKGENYEKLWVGEVOEKVZJEKRENVXDKGEVJELOMLZJLKVPJNIMV
XENXGDVXWKDOGEKQEKGDVGEKQEKDOXWKAWSNIEKGENQDKKKLZJXLNNEK

Wende das oben genannte Werkzeug an, um sie zu analysieren und zu entschlüsseln.

7.2.3 Vigenère

Im 16. Jahrhundert wurde von Blaise de Vigenère ein Verschlüsselungsverfahren erdacht, das als eine Weiterentwicklung des Caeser-Verfahrens angesehen werden kann. Die Kernidee ist, nicht für den gesamten Text dieselbe Verschiebung zu verwenden. Während wir bei Casear nur einen einzigen Buchstaben als Schlüssel verwendet haben, verwenden wir bei Vigenère ein Wort oder sogar einen Satz.

Ein sehr nützliches Werkzeug ist dabei das folgende Vigenère-Quadrat:

Klartext	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Schlüsseltext (A)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Schlüsseltext (B)	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
Schlüsseltext (C)	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
Schlüsseltext (D)	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
Schlüsseltext (E)	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
Schlüsseltext (F)	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
Schlüsseltext (G)	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
Schlüsseltext (H)	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
Schlüsseltext (I)	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
Schlüsseltext (J)	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
Schlüsseltext (K)	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
Schlüsseltext (L)	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
Schlüsseltext (M)	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
Schlüsseltext (N)	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
Schlüsseltext (O)	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
Schlüsseltext (P)	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Schlüsseltext (Q)	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Schlüsseltext (R)	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
Schlüsseltext (S)	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
Schlüsseltext (T)	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
Schlüsseltext (U)	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
Schlüsseltext (V)	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
Schlüsseltext (W)	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
Schlüsseltext (X)	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Schlüsseltext (Y)	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Schlüsseltext (Z)	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Sehen wir uns ein Beispiel an. Der Klartext soll lauten DIESER SATZ IST GEHEIM und als Schlüssel verwenden wir INFO. Zur Verschlüsselung können wir nun eine kleine Tabelle anfertigen:

Plain	D	I	E	S	E	R	S	A	T	Z	I	S	T	G	E	H	E	I	M
Key	I	N	F	O	I	N	F	O	I	N	F	O	I	N	F	O	I	N	F
Cipher	L	V	J	G	M	E	X	O	B	M	N	G	B	T	J	V	M	V	R

Wir sehen, dass der Schlüssel so oft wiederholt wird, bis man genug Einträge in der Tabelle ausgefüllt hat. Anschließend finden im Grunde Verschlüsselungen nach dem Caesar-Verfahren mit wechselnden Schlüsseln statt; das D wird mit dem Schlüssel I verschlüsselt, das folgende I mit N und so fort.

Aufgabe 32. *Verschlüssle erneut den Satz DIESER SATZ IST GEHEIM, allerdings mit dem Schlüssel GEHEIM.*

Aufgabe 33. Gegeben ist der folgende Text:

Ial qmca moh hvyek Xquylp bevwe
Gbr bbu uca ext brp tyfqimzpas
Moh kmqnml hixsi hnuhqrm Nmhk
Qrzlfqn yiplbjl tkly ugk iaay
Mba Gmlpgtt Aefkkhjf nuh Gxzgtivr
If lahxu Tanzi dxy Bhrzmw
Ciddblrfe tiunxu sumlr Lhor
Wvgt issqtssmoh ded whw aesie tbw
Xpr nxbid Alvd uldag kee Ahye
Kid lwvmca Moh nin Borqn omqrfpx zn ausllr
Lpi ltzwqn siudxy sagg hektmesxu

Pix yzufneqnzsmohx Tdaxgmeihu
Dbl mnzlrmealvfe Zgtltttugdlmf
Wpi gxox mby pal qgsl moh zgtog wmgxu lu diut
Wue rsqnglr mby paalv nbjlf plmfek hueglr
Ugk wukg gnw kgt pgt dbizdbni Iauiz
Omqrfpx znt Qrlaiz wlw fhskqnwlr Jtovqs
Moh ded zhrl lwvmcasse wvgt lv iak ie
Brp yprpeg Wue tmoh tsdgxuw mts
Ekoeqnza mn lmzef Mztxnvml
Eoh yipeg lmt qe kxprqn Gaqcd
Eo nmzg kid Khyy ulxduxix hbuaqg

Er wurde mithilfe des Vigenère-Verfahrens verschlüsselt und soll nun geknackt werden. Dazu können auch Hilfsmittel wie zum Beispiel

- `http://einklich.net/etc/vigenere.htm`
- `http://www.kas-bc.de/krypto/analyse.php`

verwendet werden.

(a) Gehen wir zunächst davon aus, dass wir keinerlei Informationen über den Text und den Schlüssel haben.
Analysiere den Text und versuche ihn zu entschlüsseln. Erläutere alle Ansätze, die Dir in den Sinn kommen und erkläre auch, ob sie funktionieren.

(b) Gehen wir nun davon aus, dass uns verraten wurde, dass der Schlüssel aus vier Zeichen besteht.
Analysiere nun mit dieser zusätzlichen Information den Text und versuche weiter, ihn zu entschlüsseln. Erläutere wieder alle Ansätze, die Dir in den Sinn kommen.

(c) Nehmen wir nun an, wir wissen, dass schon einmal mit demselben Schlüssel verschlüsselt wurde: DER EUKLIDISCHE RAUM wurde zu PEK IGKEPHUSVOI RTBG verschlüsselt.
Analysiere weiterhin mit dieser letzten Information den Text und versuche weiter, ihn zu entschlüsseln. Erläutere nach wie vor alle Ansätze, die Dir in den Sinn kommen.

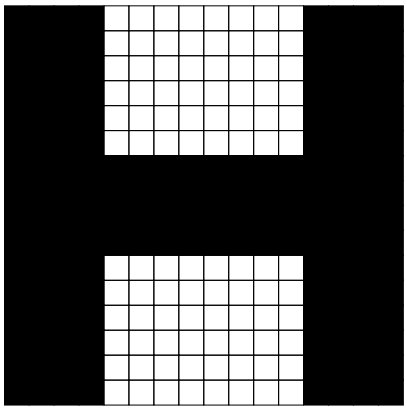
Aufgabe 34. *Beurteile die Sicherheit des Vigenère-Verfahrens — auch in Vergleich zu Caesar oder einem einfachen Substitutionsverfahren. Erläutere mögliche Schwachstellen oder mögliche Gefahren bei fahrlässiger Anwendung.*

7.2.4 One Time Pad

Die Grundidee des One Time Pad-Verfahrens ist, dass Alice und Bob sich zunächst auf einen gemeinsamen Schlüssel einigen, den sie **nur ein einziges Mal** verwenden. Der Schlüssel hat dabei dieselbe Länge wie die Nachricht, die geschickt werden soll.

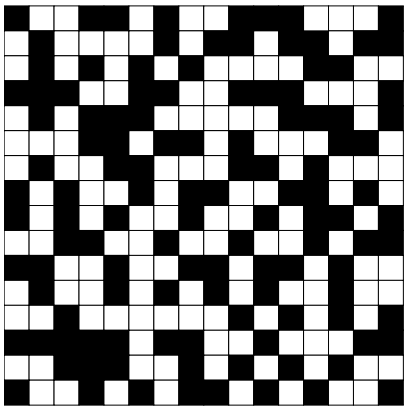
Mit diesem Verfahren könnte man wie gewohnt Text oder Zahlenfolgen verschlüsseln. Oftmals findet man in der Literatur auch eine Erklärung, in der Bitfolgen (also Folgen von Nullen und Einsen) verschlüsselt werden. Die Arbeitsweise lässt sich aber auch sehr schön Veranschaulichen, wenn man die Nachrichten bildlich darstellt. Statt Nullen und Einsen könnten wir nämlich auch schwarze und weiße Pixel verschicken. Dies ist letztlich dasselbe; man könnte eine Null durch einen weißen und eine Eins durch einen schwarzen Pixel darstellen.

Nehmen wir als Beispiel einmal diese 16×16-Pixel große Nachricht:



Hier wurde mit den Pixeln eine recht einfache Grafik erzeugt, damit man das Verfahren gut nachvollziehen kann.

Der Schlüssel ist nun ebenfalls ein 16×16-Pixel großes Feld von Schwarz und Weiß (bzw. Einsen und Nullen):



Wie eingangs erwähnt steht dieser Schlüssel Alice und Bob zur einmaligen Anwendung zur Verfügung.

Die Nachricht und der Schlüssel werden nun übereinandergelegt und in der folgenden Weise miteinander kombiniert:

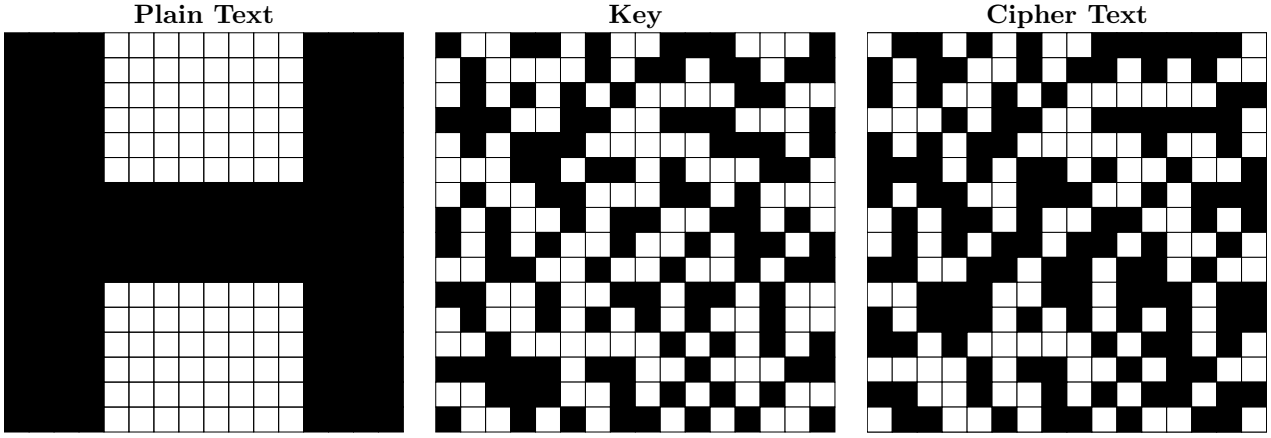
Plain Text	**Key**	**Cipher Text**
Weiß	Weiß	Weiß
Weiß	Schwarz	Schwarz
Schwarz	Weiß	Schwarz
Schwarz	Schwarz	Weiß

Interpretieren wir Weiß als 0 und Schwarz als 1, erhalten wir übrigens eine Tabelle, die uns aus einem ganz anderen Zusammenhang bekannt vorkommen sollte:

Plain Text	**Key**	**Cipher Text**
0	0	0
0	1	1
1	0	1
1	1	0

Aufgabe 35. *Erkläre, woher wir eine solche Tabelle schon kennen.*

Im folgenden Bild wird die Verschlüsselung in unserem Beispiel dargestellt:



Als Faustregel kann man sich merken: Wo im Schlüssel ein schwarzer Pixel ist, wird die Nachricht invertiert, wo im Schlüssel ein weißer Pixel ist, ändert sich nichts[9].

Aufgabe 36. Eve könnte auf die Idee kommen, nach dem Abfangen der verschlüsselten Nachricht, einfach mit Rechnerhilfe alle erdenklichen Schlüssel zu testen. Diese Möglichkeit soll nun analysiert werden. Dazu nehmen wir an, dass ihr Rechner in einer Sekunde 1000 Schlüssel testen kann. D.h. in einer Sekunde kann er 1000 verschiedene Schlüssel mit dem Cipher Text kombinieren, um so die potentielle Nachricht zu erhalten.

[9] Aufgrund der Symmetrie der Verschlüsselungsvorschrift könnte man sich als Faustregel auch merken: Wo in der Nachricht ein schwarzer Pixel ist, wird der Schlüssel invertiert, wo in der Nachricht ein weißer Pixel ist, ändert sich nichts

(a) *Berechne, wie viele Schlüssel der Rechner pro Minute, pro Stunde, pro Tag und pro Jahr testen kann.*

(b) *Berechne, wie viele Schlüssel bei einem Feld von* 16×16 *Pixeln Alice und Bob zur Auswahl stehen.*

(c) *Berechne, wie lange Eve warten muss, bis alle Schlüssel getestet wurden.*

Aufgabe 37. *Beurteile die Sicherheit des One Time Pad-Verfahrens unter der Voraussetzung, dass es korrekt angewendet wird (d.h. dass ein Schlüssel nur einmal verwendet wird).*

Aufgabe 38. (a) *Ermittle, welchen Cipher Text man erhält, wenn man eine Nachricht mit einem komplett weißen Schlüssel verschlüsselt.*

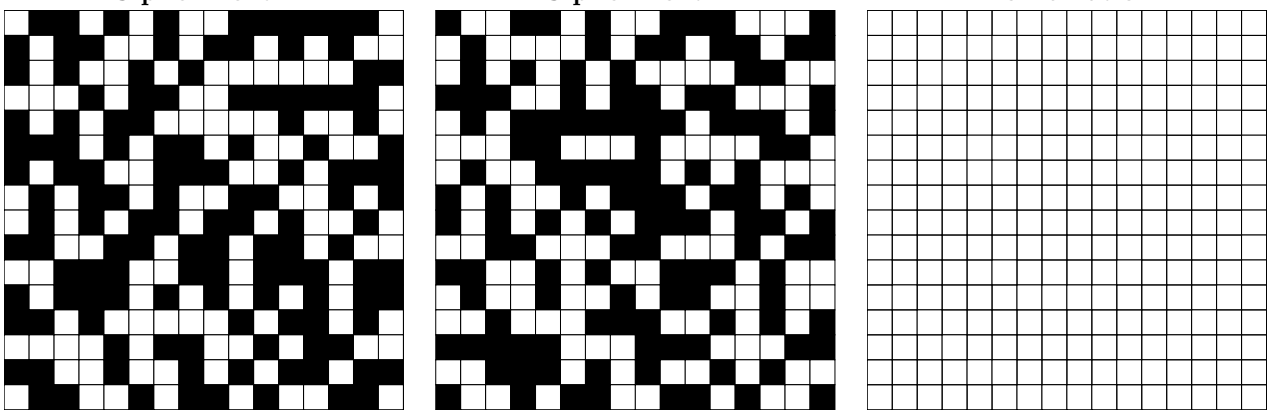
(b) *Ermittle, welchen Cipher Text man erhält, wenn man eine Nachricht mit einem komplett schwarzen Schlüssel verschlüsselt.*

(c) *Ermittle, welchen Cipher Text man erhält, wenn man eine Nachricht mit sich selbst als Schlüssel verschlüsselt.*

Es wurde schon mehrfach betont, dass ein Schlüssel nur einmal verwendet werden darf. Warum das so wichtig ist, ist Gegensand der folgenden Aufgabe.

Aufgabe 39. Nehmen wir an, Eve hat zwei verschlüsselte Nachrichten abgefangen. Sie vermutet, dass Alice und Bob den Fehler begangen haben, zweimal denselben Schlüssel zu verwenden. Ihr Ansatz ist nun, die beiden Nachrichten mithilfe der bekannten Methode zu kombinieren. D.h., sie verwendet einen Cipher Text, um den anderen Cipher Text damit zu entschlüsseln.

Cipher Text 1 **Cipher Text 2** **Kombination**



Ermittle Eves Ergebnis und ermittle damit, welche Nachrichten geschickt wurden.

Aufgabe 40. *Erläutere (mit mathematischen oder logischen Argumenten), warum sich bei der Kombination in Aufgabe 39 das dort zu sehende Bild ergeben hat. Mit anderen Worten, erkläre, wodurch die Angreifbarkeit bei zweifacher Verwendung desselben Schlüssels entsteht.*

Aufgabe 41. *Beurteile die praktische Einsetzbarkeit des One Time Pad-Verfahrens.*

7.3 Asymmetrische Verfahren

7.3.1 Variation von Caesar

Eingangs wurde bereits erläutert, dass man bei asymmetrischen Kryptosystemen zum Ver- und Entschlüsseln verschiedene Schlüssel benötigt. Außerdem darf es nicht möglich sein, mithilfe des Schlüssels zum Verschlüsseln den fehlenden Schlüssel zum Entschlüsseln zu ermitteln.

Um einen ersten Eindruck davon zu erhalten, wie es sein kann, dass man zum Ver- und Entschlüsseln verschiedene Schlüssel benötigt, schauen wir eine Variante des Caesar-Verfahrens an. **Achtung, diese stellt noch kein vollwertiges asymmetrisches Verfahren dar, denn man kann aus dem Schlüssel zum Verschlüsseln den anderen berechnen.** Ein vollwertiges Verfahren folgt später.

Beim Caesar-Verfahren haben wir Funktionen der Gestalt

$$f_a(x) = x + a \mod 26$$

zum Verschlüsseln verwendet. Wir könnten dies nun variieren, indem wir statt der Addition die Multiplikation verwenden:

$$f_a(x) = x \cdot a \mod 26$$

Beispielweise könnten wir mit $a = 3$ eine Tabelle zum Verschlüsseln erstellen, indem wir $f_3(0) = 0$, $f_3(1) = 3$, $f_3(2) = 6$,…berechnen.

Aufgabe 42. *Berechne die fehlenden Einträge der Verschlüsselungstabelle:*

Klartext	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Schlüsseltext	0	3	6																							

Zusatzaufgabe 1. Nicht jeder Wert von a führt in diesem Verfahren zu einer sinnvollen Verschlüsselung. *Analysiere, welche Eigenschaft a haben muss, um zu einer brauchbaren Verschlüsselung zu führen.*

Nun stellt sich die Frage, wie die Entschlüsselungsfunktion g_3 aussehen könnte.

Aufgabe 43. Ein erster Vorschlag könnte sein, dass man $g_3(x) = \frac{x}{3} \mod 26$ verwendet. Beispielsweise würde ja dann $g_3(6) = 2$ gelten, so dass die Entschlüsselung scheinbar klappt.
Erläutere, warum diese Funktion dennoch nicht in Frage kommt.

Wie in der letzten Aufgabe gesehen, kann man mit dem Schlüssel zum Verschlüsseln (in diesem Fall 3) für gewöhnlich nicht Entschlüsseln. Einen geeigneten Schlüssel d zum Entschlüsseln kann man mithilfe der Bedingung

$$3 \cdot d \mod 26 = 1$$

finden. D.h. man muss ein Zahl d finden, mit der Eigenschaft, dass $3 \cdot d$ bei der Division mit 26 einen Rest von 1 liefert.

Aufgabe 44. (a) *Ermittle eine geeignete Zahl d.*
Hinweis: Wir haben dazu kein systematisches Vorgehen besprochen. Im schlimmsten Fall muss man hier also einfach ausprobieren.

(b) *Ermittle die zugehörige Funktion g_d zum Entschlüsseln und teste anhand von Rechenbeispielen, ob diese Funktion tatsächlich funktioniert.*

7.3.2 RSA

Kommen wir nun zu dem ersten in der Praxis verwendeten asymmetrischen Verschlüsselungsverfahren, dem RSA-Verfahren, das 1977 von Rives, Shamir und Adleman entwickelt wurde.

Wie bereits mehrfach betont gibt es hier zwei Schlüssel. Üblicherweise werden diese *privater* und *öffentlicher* Schlüssel genannt:

privater Schlüssel	**öffentlicher Schlüssel**
kennt nur Bob	darf jeder kennen — damit natürlich auch Alice
wird zum Entschlüsseln gebraucht	wird zum Verschlüsseln gebraucht
lässt sich nicht aus dem öffentlichen Schlüssel berechnen	

Bei diesem Verfahren werden einige mathematische Fakten und Regeln verwendet, die wir leider nicht alle im Detail besprechen können. Daher sollten wir diesen Abschnitt mit dem Ziel betrachten, dass wir die Arbeitsweise grob verstehen wollen.

Damit Alice eine Nachricht an Bob schicken kann, muss Bob die beiden Schlüssel erzeugen.

(1) Zunächst benötigt er zwei große Primzahlen p und q. Diese muss er streng geheim halten! Er berechnet dann $N = p \cdot q$. Dieses N veröffentlicht er.
Um die Rechnungen nachzuvollziehen verwenden wir ganz kleine Primzahlen; $p = 5$, $q = 11$. Dann haben wir $N = 55$.

(2) Nun berechnet Bob die sogenannte phi-Funktion von N:

$$\varphi(N) = (p-1) \cdot (q-1)$$

In unserem Beispiel $\varphi(55) = 4 \cdot 10 = 40$. Auch dieser Wert ist streng geheim!

(3) Bob wählt nun vollkommen frei eine Zahl e zwischen 1 und $\varphi(N)$ mit der Eigenschaft $\text{ggt}(e, \varphi(N)) = 1$. Diese Zahl veröffentlicht er. Damit ist der öffentliche Schlüssel fertig; er besteht aus N und e.
Wir könnten $e = 13$ wählen, denn $\text{ggt}(13, 40) = 1$. Man muss für e nicht zwangsläufig eine Primzahl nehmen. 21 wäre zum Beispiel auch gegangen.

(4) Nun benötigt er noch eine Zahl d mit der Eigenschaft $e \cdot d \mod \varphi(N) = 1$. Es gibt ein Verfahren, um sie systematisch zu berechnen, aber dieses soll nun nicht näher betrachtet werden.
In unserem Fall funktioniert $d = 37$, denn $13 \cdot 37 = 481$ und $481 \mod 40 = 1$.

Damit ist die Produktion der Schlüssel beendet:

privater Schlüssel	**öffentlicher Schlüssel**
$d = 37$	$N = 55$, $e = 13$

Jetzt kann Alice eine Nachricht verschlüsseln und an Bob schicken:

Verschlüsseln. Alice hat eine Nachricht $0 < m < N$. Sie berechnet $c = m^e \mod N$ und schickt c an Bob.
In unserem Beispiel könnte sie $m = 24$ zu $c = 24^{13} \mod 55 = 19$ verschlüsseln.

Entschlüsseln. Bob berechnet $c^d \mod 55$ und erhält damit wieder m.
Im Beispiel gilt tatsächlich $19^{37} \mod 55 = 24$.

Aufgabe 45. Stellen wir uns vor, wir wären in der Rolle des Angreifers Eve. Wir kennen dann den öffentlichen Schlüssel (also e und N) und haben die verschlüsselte Nachricht c abgefangen.
Analysiere, welche Schritte nun notwendig wären, um die Nachricht zu entschlüsseln. Analysiere, inwiefern die Größe der Primzahlen p und q dabei eine Rolle spielt.

Wir wollen nun ein gerne auftretendes Missverständnis[10] bei der Beschreibung des RSA-Verfahrens betrachten. Nehmen wir an, wir wollen die recht kurze Nachricht RSA verschlüsseln. Dann sollten wir zunächst die drei Buchstaben in Zahlenwerte umwandeln, wie wir es bereits beim Caesar-Verfahren gesehen haben. Aus RSA wird dann 171800. Diese Zahl gilt es nun zu verschlüsseln. Wir könnten nun die beiden Primzahlen $p = 971$ und $q = 997$ wählen, damit $N = 968087$ berechnen und den öffentlichen Schlüssel $e = 1331$ wählen. Die Nachricht wird dann verschlüsselt zu $171800^{1331} \mod 966120$.

Was wir **keinesfalls** tun sollten, ist die Buchstaben einzeln zu verschlüsseln. D.h., wir sollten nicht $17^{1331} \mod 966120$, $18^{1331} \mod 966120$ und $00^{1331} \mod 966120$ berechnen und die drei Ergebnisse senden. Noch viel weniger sollten wir bei längeren Nachrichten die Buchstaben einzeln verschlüsseln.

Aufgabe 46. *Erläutere, warum man so auf keinen Fall vorgehen sollte.*

[10] Dieses trat leider auch in einen ansonsten sehr ordentlich recherchierten Facharbeit auf.