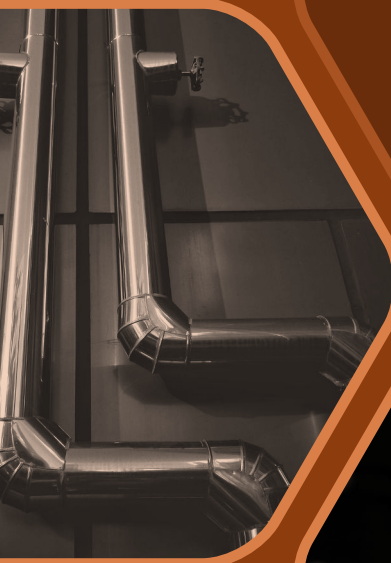




LESEPROBE · INF.SCHOOL · EINSATZ IM UNTERRICHT NUR MIT LIZENZ

Computernetzwerke und das Internet

Von Röhren, Paketen und Matrjoschkas

FÜR DEN INFORMATIKUNTERRICHT AM GYMNASIUM

INF school

Inhaltsverzeichnis

1	Einleitung	5
2	Willkommen auf den Pipe Islands	7
	Von Schichten und Protokollen	9
3	Die Netzzugangsschicht (Link Layer)	13
	Topologie eines Netzwerkes	13
	Ethernet	16
4	Die Internetschicht (Internet Layer)	21
	Rohrpostadressen	22
	IP: das Internetprotokoll	23
	IP-Adressen	23
	Netzmasken	24
	IP-Pakete	26
	Nah oder fern?	26
	Pakete in der Rohrpost	28
	Die Reise eines IP-Pakets	29
	Hierarchische Adressen	34
	Router: die Verteilzentren des Internets	35
	RIP: Routingtabellen erlernen	40
5	Die Transportschicht (Transport Layer)	45
	Ports	46
	UDP: Keine Haftung für verlorene Nachrichten.	47
	TCP: Nachrichten per Einschreiben	48
	Vom Datenstrom zu TCP-Segmenten	50
6	Die Applikationsschicht (Application Layer)	55
	Webseiten und Domainnamen	56
	DNS – Namen werden zu IP-Adressen	57
	HTTP – Webseiten anfordern	58
	Eine Webseite auf Reisen	60
7	Lösungsvorschläge	63

Über dieses Skript

Es freut mich sehr, dass du¹ dich entschieden hast, dieses Lehrmittel zu verwenden. Ich hoffe, dass es dich auf deinem Weg zur gymnasialen Matura begleiten und unterstützen wird, indem es dein Interesse am Fachgebiet Informatik weckt, dich zur selbstständigen Erkundung der angesprochenen Themengebiete motiviert, und dir die Grundlagen bietet, deinen Weg in der digitalen Zukunft zu beschreiten.

Dieses Skript führt dich in die Grundlagen von Computernetzwerken und des Internets ein. Anhand einer fiktiven Rohrpost-Inselwelt erkunden wir, wie Daten ihren Weg vom Sender zum Empfänger finden, welche Aufgaben die verschiedenen Schichten übernehmen und wie deren Protokolle zusammenspielen.

Die Flughöhe versuche ich so hoch wie möglich, und so niedrig wie nötig zu halten. Ein intuitives Verständnis der Materie ist mir wichtig, genauso aber soll man wichtige Konzepte im Detail anschauen und beleuchten. Zudem sollen für interessierte SuS weiterführende Informationen für die eigenständige Vertiefung zur Verfügung gestellt werden.

Feedback

Egal ob du zu den Lehrpersonen oder zu den Lernenden gehörst: Wenn du Fehler oder Ungereimtheiten findest, Verbesserungsvorschläge hast, oder andere Ideen hast, wie man dieses Lehrmittel verbessern könnte, dann melde dich ungeniert bei mir! Ein kurzes Mail an feedback@koehlf.ch genügt. Davon lebt dieses Skript, dadurch wird es besser, und du kriegst nebenbei auch einen Eintrag in der *List of bug hunters*.

Danke!

Ich möchte allen Lehrpersonen und SchülerInnen einen grossen Dank aussprechen, die es durch ihr Feedback ermöglichen, dieses Skript ständig zu verbessern.

Ein grosses Dankeschön geht auch an die ganze $\text{\LaTeX}$ -Community auf StackExchange, ohne welche ich dieses Skript niemals in dieser Form hätte schreiben können.

Zu guter Letzt möchte ich mich bei meinen Eltern bedanken, die mich gross² gezogen haben, mich jederzeit in all meinen Zielen unterstützt haben und immer an mich geglaubt haben. Tusen takk! ♥

©2026 Florian Köhl

¹bzw. deine Lehrperson

²sehr gross

Einleitung

Eine Welt ohne Internet können wir uns heute kaum mehr vorstellen. Vom Smartphone über den Drucker bis zum Toaster, alles ist heute irgendwie miteinander vernetzt. Beinahe jedes Cafe ist heute ein Internetcafe, da es 'Free WiFi' anbietet. Traditionelles Fernsehen¹ ist was für alte Leute und Radio² etwas für noch ältere Leute, denn heutzutage wird alles 'gestreamt', wann und wo wir wollen. (Liebes)Briefe werden heute auch nicht mehr geschrieben, eine Messenger-Nachricht oder eine Email reicht aus. Wichtige Daten werden nicht mehr auf einer externen Festplatte gesichert, sondern wandern 'in die Cloud'. Und wer bitte schön zahlt heute noch mit Bargeld, wenn man einfach die Karte oder das Smartphone hinhalten kann³?

Doch wie funktioniert dieses Wunder namens Internet eigentlich? Wie ist es aufgebaut, und warum funktioniert es so stabil und zuverlässig, wie wir es uns gewohnt sind? Wie finden die Bits und Bytes den Weg durch dieses Labyrinth von unzähligen Kabeln und Rechnern, scheinbar ohne dass dabei jemals etwas verloren geht?

In diesem Kapitel wollen wir Antworten zu diesen Fragen finden, ohne dass wir uns zu fest in Details verrennen. Computernetzwerke sind so komplex, dass wir uns ganze zwei Jahre damit beschäftigen könnten. Auch wenn dies sicherlich äusserst interessant und spannend wäre, haben wir leider nicht so viel Zeit zur Verfügung. Wir fokussieren uns deshalb auf die groben Konzepte und Ideen, die hinter dem Begriff Internet stehen und dieses zu dem robusten Wegbegleiter machen, den wir aus unserem Alltag kennen.

Aber was hat das Internet mit Matroschkas und der Rohrpost zu tun? Auf den ersten Blick wohl nicht allzu viel, und doch werden wir sehen, dass es gewisse Gemeinsamkeiten gibt. In diesem Kapitel werden wir nämlich sehr stark mit Analogien arbeiten. Einerseits natürlich, um die trockene Theorie etwas aufzulockern, aber andererseits - und das ist die eigentliche

¹Dieses komische viereckige Gerät mit Bildschirm und einer Antenne

²Das andere viereckige Gerät mit Lautsprecher und Antenne

³Ok, auch hier gibt es Ausnahmen bzw. Relikte aus vergangenen Zeiten, zum Beispiel Snackautomaten in Klosterschulen. Oder das Grossväterchen, welches den Tageseinkauf mit 5-Räpplern zahlt und immer, aber wirklich immer genau vor einem an der Kasse ansteht.

011100110110001101101000011011 10110111101101100

Absicht dahinter - um das Zusammenspiel der einzelnen Komponenten in einem Netzwerk besser zu verstehen, indem wir eine Brücke zu uns bekannten Konzepten schlagen können.

Aber Achtung: Keine Analogie deckt sich hundertprozentig mit der Realität! Es handelt sich immer um Vergleiche, die Konzepte in einem anderen Kontext aufzeigen sollen. Je detaillierter man eine Analogie auseinandernimmt, desto weniger nützlich ist sie, und kann schlussendlich mehr Verwirrung stiften als Klarheit schaffen.

Nach diesem Wort der Vorsicht wollen wir uns jetzt ins Abenteuer stürzen. Wir begeben uns dafür auf eine Reise in eine (fiktive) Inselwelt...

Willkommen auf den Pipe Islands

Reto und Trudi leben auf den Pipe Islands, einer kleinen abgeschotteten Inselgruppe irgendwo im informatischen Ozean. Das spezielle an diesen Inseln ist ihre Art der Kommunikation, sie setzen nämlich ausschliesslich auf die Rohrpost. Jedes Gebäude ist an das Rohrnetzwerk angeschlossen, und jegliche Kommunikation verläuft ausschliesslich über dieses Medium. Die Rohrpostleitungen bestehen innen aus zwei getrennten Richtungsrohren, so dass in beide Richtungen Dosen verschickt werden können, ohne dass diese miteinander kollidieren.



Abbildung 2.1: Tubingen (links) und Rohrbach (rechts)

Es ist Mittagszeit. Trudi sitzt in ihrem Büro im obersten Stock des violetten Gebäudes in Tubingen. Da sie einen Anflug von Hunger verspürt, entschliesst sie sich, sich ein Mittagessen

liefern zu lassen. Reto's Restaurant in Rohrbach ist auf den ganzen Pipe Islands für seine schmackhaften Mittagsmenus bekannt, deshalb fällt ihr die Wahl leicht. Sie bestellt einen Salat als Vorspeise, die Pasta des Tages als Hauptgang und ein feines Tiramisu zum Dessert.

Wir wollen mal kurz veranschaulichen, wie so eine Bestellung auf den Pipe Islands abläuft. Trudi schreibt ihre Bestellung auf ein Blatt Papier, dieses landet dann irgendwie in einer Rohrpostbüchse, und diese wiederum wird dann in Trudis Gebäude in das Röhrennetzwerk eingespiesen. Von da aus muss die Bestellung dann erst mal den Weg nach Rohrbach finden, und dort angekommen dann zu Reto's Restaurant umgeleitet werden, wo Reto schlussendlich die Bestellung entgegen nehmen kann.

Reto macht sich natürlich sofort an die Arbeit und bereitet die bestellten Gerichte zu. Leider haben nicht alle drei Gänge in einer Rohrpostbüchse Platz, deshalb muss er die Bestellung auf drei Büchsen aufteilen. Gesagt, getan - und ab geht die (Rohr)Post, dieses mal müssen aber drei verschiedene Büchsen ihren Weg zu Trudis Büro finden, damit diese ihren Hunger stillen kann.

Diese Art von Kommunikation ist auch in der Computerwelt weit verbreitet. Natürlich meinen wir hier nicht die Transportmethode der Rohrpost und auch nicht die Bestellung eines Mittagessens, sondern ganz allgemein die Situation, in der jemand etwas bestellt, und jemand anderes diese Bestellung bearbeitet und beantwortet. Dieses Modell nennt man auch **Client-Server Modell**

CLIENT-SERVER-MODELL



Das Client-Server-Modell ist eine Möglichkeit der Aufgabenverteilung in einem Netzwerk. Die beiden Hauptrollen darin sind, wie der Name schon vermuten lässt, der Client und der Server.

Ein **Client** ist ein Programm oder ein Prozess, welcher einen Dienst (oft engl. **Service**) in Anspruch nehmen möchte. Ein Browser beispielsweise möchte den Inhalt einer Webseite anfordern, um sie dem Benutzer anzeigen zu können.

Ein **Server** (von engl. to serve) ist ein Programm oder ein Prozess, welcher diesen Service bereitstellt. Ein Webserver etwa könnte die obige Anfrage beantworten und den Inhalt der angeforderten Webseite liefern, oder "servieren".

Die Anfrage eines Clients an einen Server nennt man **Request**, die Antwort, welche der Server zurück schickt, heisst **Response**. Wenn diese Unterscheidung nicht relevant ist, so spricht man meistens von **Nachrichten** (engl. messages), welche über ein Netzwerk ausgetauscht werden.

In unserem Beispiel oben wäre also Trudi der Client (die Klientin?), und Reto der Server. Beispiele aus der realen Welt gibt es zuhauf. Wenn du eine Datei in deinen Online-Speicher hochlädst, benutzt du einen Service. Mailserver bieten meist gleich zwei Services an: einen, um Mails zu versenden, und einen, um Mails zu empfangen. Und Fans von Multiplayer-Spielen sind sicherlich auch mit dem Begriff Server vertraut.

Doch zurück zu Trudis Essensbestellung. Wir wollen etwas genauer untersuchen, was hinter den Kulissen alles abläuft, wenn Trudi einen Request an Reto bzw. wenn dieser seine Response zurück schickt. Dazu überlegen wir uns mal, was denn alles schief laufen könnte, wenn solche

Nachrichten über das Netzwerk geschickt werden. ...



AUFGABE 1: DELIVERY ISSUES

- Was für Probleme könnten bei einer solchen Essensbestellung, also dem Austausch von Nachrichten in einem Netzwerk, auftreten? Zähle mindestens drei auf.
- Hast du schon eine Idee, wie diese Probleme verhindert oder gelöst werden könnten?

Sicherlich hast du einige Punkte gefunden, die Trudi bzw. ihrem Mittagessen einen Strich durch die Rechnung machen könnten. Ein paar Beispiele wollen wir hier aufzählen.

- Wegen eines Rohrschadens gehen einzelne **Nachrichten verloren**, die durch diesen Röhrenabschnitt geschickt werden. In unserer Analogie würde so vielleicht die Pasta irgendwo auf dem Meeresgrund landen.
- Eine Nachricht wird beim Transfer in einem Rohrabschnitt **beschädigt**: Das feine Tiramisu kommt völlig zerquetscht an.
- Eine Nachricht wird **an den falschen Ort** geliefert. Stell dir vor, Trudis Bestellung landet statt bei Reto's Restaurant beim Aussichtsturm in Tübingen oder auf einer völlig falschen Insel...
- Eine Nachricht kommt **nicht in der korrekten Reihenfolge** an, da die einzelnen Teile unterschiedliche Wege durch das Röhrennetzwerk nehmen. Dann müsste Trudi vielleicht erst das Tiramisu, dann den Salat und erst dann die Pasta essen¹.
- Ein wichtiges **Verbindungsrohr** zwischen Tübingen und Rohrbach **ist defekt**, die Inseln sind nun völlig voneinander abgeschnitten, oder die Inseln sind nur noch durch Umwege miteinander verbunden.

Du siehst, es kann also so einiges schiefgehen. Stell dir vor, Trudi und Reto müssten sich selbst um all diese Probleme kümmern bzw. sicher stellen, dass diese nicht auftreten. Da würde ja gar keine Zeit mehr für ihre eigentlichen Aufgaben bleiben! Das gleiche gilt natürlich für Computernetzwerke. Wenn jede Applikation ihre Nachrichten selbst sicher ans Ziel führen müsste, wäre das Entwickeln solcher Programme eine Herkulesaufgabe!

Von Schichten und Protokollen

Das Schichtenmodell

Um solche komplexe Situationen zu vereinfachen, greifen wir in der Informatik oft auf das Mittel der **Abstraktion** zurück. In der Netzwerkwelt wird die Kommunikation zwischen zwei Geräten in verschiedenen Ebenen oder **Schichten** (engl. **layers**) organisiert. Jede Ebene ist

¹Zugegeben, es gibt schlimmeres als Tiramisu als Vorspeise, aber es geht hier ums Prinzip :-)

für bestimmte Aufgaben zuständig und löst gewisse möglicherweise auftretende Probleme. Ein solches Modell nennt man **Schichtenmodell**. Jenes Modell, welches wir hier genauer anschauen können und welches uns im Rest dieses Kapitels begleitet, ist das **TCP/IP** Schichtenmodell. Dieses definiert 4 Schichten:



Jede Ebene baut auf den darunter liegenden Schichten auf und kann auf diese zurückgreifen, um ihre spezifischen Aufgaben zu erledigen. Doch welche Aufgaben sind dies genau?

Die **Anwendungsschicht** ist zuständig für die Übertragung von Daten zwischen zwei Applikationsprozessen.

Die Aufgabe der **Transportschicht** ist es, einen Kommunikationskanal für den Datenaustausch von zwei Applikationen bereit zu stellen. Diese Schicht ist auch für die Vollständigkeit und die korrekte Übermittlungsreihenfolge der Nachrichten zuständig, falls dies denn gewünscht ist.

Die **Internetschicht** ist für die Vermittlung von Paketen **zwischen verschiedenen** Netzwerken zuständig. Ihre Aufgabe ist es also, einen Weg vom Ursprungs- zum Zielnetzwerk zu finden.

Die **Netzzugangsschicht** sorgt für die Übertragung der Nachricht **innerhalb eines Netzwerkes**, also zwischen zwei Geräten, die sich im gleichen Netzwerk befinden. Dabei wird davon ausgegangen, dass ein (physikalisches) Medium zur Übertragung der Daten vorhanden ist. In manchen Modellen wird dieser physikalische Aspekt als eigene Schicht betrachtet und wird dann **Bitübertragungsschicht** (engl. Physical Layer) genannt.

Wie sieht das nun auf den Pipe Islands aus? Trudi und Reto arbeiten auf der **Büroebene**, also jeweils im 4. Stock ihrer Gebäude. Sie möchten miteinander kommunizieren bzw. Nachrichten austauschen. Sie tun dies aber nicht direkt untereinander, sondern über die Abteilungen, die sich in den darunterliegenden Stockwerken befinden. Trudi gibt zum Beispiel ihre Bestellung also erst einmal eine Etage tiefer in die Transportabteilung weiter, für sie ist die Sache damit vorerst erledigt.

Die **Transportabteilung** befindet sich im dritten Stock und hat zwei Hauptaufgaben: Einerseits sorgt sie dafür, dass eingehende Nachrichten (sei es eine Bestellung oder das tatsächliche Essen) im richtigen Büro landen. Das bestellte Essen soll ja schliesslich zu Trudi ins Büro, und nicht in die Werkstatt. Weiter ist die Transportabteilung dafür verantwortlich, dass Nachrichten auch vollständig und in der richtigen Reihenfolge an der Destination ankommen, sofern diese Garantie erwünscht ist. Wenn Trudis Essen also geliefert wird, dann leitet die

Transportabteilung dieses erst an Trudi weiter, wenn sie das in der richtigen Reihenfolge tun kann. Wie genau das Essen aber von einem Ort zum anderen gelangt ist nicht Aufgabe der Transportabteilung, sondern der darunter liegenden Verteilabteilung.

Die **Verteilabteilung** befinden sich jeweils in den zweiten Etagen der Gebäude und sind dafür verantwortlich, dass die Nachrichten den Weg zwischen den verschiedenen Inseln finden. Dazu arbeitet sie mit den verschiedenen Verteilzentren auf dem Weg zusammen.

Auf dem untersten Stockwerk befinden sich jeweils die **Rohrpostabteilungen**. Diese sind dafür verantwortlich, die Nachrichten von Gebäude zu Gebäude zu verschicken - via Rohrpost.

Protokolle

Mit der Unterteilung in Schichten allein ist das Problem aber noch nicht gelöst. Wichtig ist, dass diese Schichten untereinander kommunizieren können! Wenn in Rohrbach der Absender am linken Ende der Dose geschrieben wird und der Empfänger auf der rechten, in Tübingen aber genau umgekehrt, so würden die Dose endlos zwischen den beiden Inseln hin und her geschickt. Wenn Trudi ihre Bestellung aufgibt, muss sie wohl ein bestimmtes Bestellformular ausfüllen, welches Reto ebenfalls kennt.

In der realen Welt ist dies ebenfalls der Fall: Geräte unterschiedlicher Hersteller müssen zusammenarbeiten, um den Austausch von Daten zwischen Rechnern zu ermöglichen. Es braucht deshalb eine weitere Form der Standardisierung, um das reibungslose Zusammenspiel zwischen Rechnern oder Netzwerken (bzw. Gebäuden und Inseln) zu garantieren. Solche Abmachungen nennt man Kommunikationsprotokolle, oder kurz **Protokolle**.

KOMMUNIKATIONSPROTOKOLL



Unter einem Kommunikationsprotokoll versteht man eine Menge an Regeln, welche die Kommunikation zwischen zwei Kommunikationsparteien bestimmen.

Auch zwischen uns Menschen halten wir uns implizit an gewisse Protokolle, auch wenn diese nirgends genau vorgeschrieben sind. Wenn du eine Bekannte oder einen Bekannten auf der Strasse triffst, dann läuft wahrscheinlich immer nach einem gewissen Schema ab:

- *Hoi X*
- *Hoi Y*
- *Wie gahts?*
- *Guet und dier?*
- *Jaja, es lauft.*
- *Du los mal, hesh scho ghört dass ...*

Auf jeder Schicht finden sich eine Vielzahl von Protokollen, einige davon werden wir uns in diesem Kapitel etwas detaillierter anschauen. Was du dir aber im Hinterkopf behalten

solltest, ist, dass jedes Protokoll **auf einer bestimmten Schicht eingesetzt** wird, und nicht etwa zur Kommunikation zwischen zwei oder mehr Schichten verwendet wird.

Die Internetschicht (Internet Layer)

Die unterste Schicht, die Netzzugangsschicht, ist für den Versand von Nachrichten innerhalb eines lokalen Netzwerkes zuständig. Wie können wir nun Daten über diese Netzwerkgrenzen hinaus, also zwischen zwei Netzwerken, austauschen? Dies fällt ins Zuständigkeitsgebiet der nächstoberen Schicht, der **Internetschicht**, oder englisch **Internet Layer**.

Du fragst dich jetzt vielleicht, weshalb es dafür überhaupt eine eigene Schicht braucht. Können die Daten nicht einfach von Switch zu Switch weitergeleitet werden, bis sie am Ziel ankommen? Sofern der Absender die MAC-Adresse des Empfängers kennt, sollte das ja kein Problem sein... Oder etwa doch?



AUFGABE 6: ÜBERFORDERTE SWITCHES

Weshalb wären Switches mit der Aufgabe, Daten zwischen Geräten in beliebigen, unterschiedlichen Netzwerken auszutauschen, überfordert?

Erinnere dich dafür daran, wie ein Switch lernt, für welche MAC-Adresse welcher Ausgangsport benutzt werden soll und wie er sich diese Zuordnung merkt. Was geschieht ausserdem, wenn die Empfängeradresse einem Switch noch nicht bekannt ist?

Wenn wir alle Geräte der Welt zu einem einzigen riesigen lokalen Netzwerk zusammenschliessen würden, müssten sich Switches eine gewaltige Anzahl von MAC-Adressen und deren zugehörige Ausgänge merken. Noch problematischer wird es bei unbekannten Empfängeradressen: Ein Switch flutet ein solches Frame auf alle Ausgänge ausser demjenigen, von welchem es gekommen ist. In einem kleinen lokalen Netzwerk ist das kein grosses Problem. In einem weltweiten Netzwerk würde dadurch aber sehr viel unnötiger Datenverkehr entstehen.

Dazu kommt ein grundsätzlicher Nachteil der MAC-Adressen: Aus einer MAC-Adresse lässt sich nicht herauslesen, wo sich das adressierte Gerät befindet. Eine Adresse wie 3d:33:28:4d:ab:72 verrät einem Switch nicht, ob sich das Gerät im gleichen Schulhaus, in einem anderen Land oder auf einem anderen Kontinent befindet.

Du siehst also, mit MAC-Adressen und Switches allein kommen wir hier nicht weiter. Statt die Netzzugangsschicht nun immer komplizierter zu machen, führen wir eine weitere Schicht ein, die auf dem Link Layer aufbaut. Sie soll es uns ermöglichen, viele lokale Netzwerke zu einem grossen Verbund zusammenzuschliessen und Nachrichten auch über Netzwerkgrenzen hinweg zu versenden.

INTERNETSCHICHT



Der Link Layer ist dafür ausgelegt, Daten **innerhalb eines lokalen Netzwerks** zu übertragen.

Die **Internetschicht (Internet Layer)** baut darauf auf und ist dafür zuständig, Daten **über die Grenzen lokaler Netzwerke hinweg** zum richtigen Ziel zu führen.

Rohrpostadressen

Die Situation auf den Pipe Islands ist ganz ähnlich: Die Rohrpostabteilung verteilt Rohrpost innerhalb einer Insel. Für diese lokale Kommunikation reichen die Gebäudenamen, die innerhalb einer Insel eindeutig sind. Möchten wir aber Nachrichten zwischen beliebigen Inseln austauschen, wäre es unpraktisch, wenn jede Zwischenstation jeden Gebäudenamen der gesamten Inselwelt kennen müsste.

Wir führen deshalb auf einer neuen Ebene eine weitere Adressierung ein. Die Verteilabteilung arbeitet mit **Paketen**, deren Absender und Empfänger nicht mit Gebäudenamen, sondern mit einer **Rohrpostadresse** angegeben werden. Eine solche Rohrpostadresse besteht aus zwei Teilen:

- Die **Inselnummer** bezeichnet die Insel, auf der sich das Ziel befindet.
- Die **Hausnummer** bezeichnet ein bestimmtes Gebäude innerhalb dieser Insel.

Trudis Tischlerei auf Tübingen könnte beispielsweise die Rohrpostadresse

302 - 10

haben. Die Inselnummer 302 sagt uns, dass das Ziel auf Tübingen liegt, die Hausnummer 10 bezeichnet Trudis Gebäude auf dieser Insel. Retos Restaurant auf Rohrbach könnte entsprechend die Adresse 712 - 21 besitzen.

Damit haben die Gebäude auf den Pipe Islands nun zwei unterschiedliche Adressen, die für unterschiedliche Aufgaben verwendet werden:

- Der **Gebäudename** wird von der Rohrpostabteilung für die lokale Zustellung innerhalb einer Insel verwendet.
- Die **Rohrpostadresse** wird von der Verteilabteilung verwendet und enthält zusätzlich die Information, *wo* auf den Pipe Islands sich das Ziel befindet.

Genau diese Idee wollen wir nun auf Computernetzwerke übertragen.

IP: das Internetprotokoll

Das mit Abstand am häufigsten verwendete Protokoll auf der Internetschicht ist **IP**, das **Internet Protocol**. Davon gibt es zwei verbreitete Versionen: IPv4 und IPv6. Wir befassen uns hier lediglich mit der ersten, älteren Variante, werden aber später darauf zurückkommen, weshalb eine neue Version benötigt wurde.

Die Einheit von Daten, mit der IP arbeitet, heisst **IP-Paket**, oder einfach kurz Paket. Damit solche Pakete über Netzwerkgrenzen hinweg zum richtigen Ziel gelangen können, verwendet IP eine eigene Art von Adressen: die **IP-Adressen**.

IP-Adressen

Den Begriff IP-Adresse hast du bestimmt schon mal irgendwo aufgeschnappt. Technisch gesehen ist eine IPv4-Adresse nichts anderes als eine Binärzahl mit 32 Bit, also 4 Byte. Eine solche Adresse könnte beispielsweise so aussehen (die Abstände sind nur der besseren Lesbarkeit wegen dargestellt):

```
11000000 10101000 00000000 00000001
```

Da diese Darstellung für uns Menschen nicht sehr übersichtlich ist, schreibt man eine IPv4-Adresse meist in vier Gruppen à 1 Byte. Die einzelnen Bytes werden als Dezimalzahlen dargestellt und durch Punkte getrennt. Diese Schreibweise nennt man **dotted decimal notation** oder zu Deutsch **Dezimalpunktschreibweise**. Die obige IP-Adresse würde damit als

```
192.168.0.1
```

geschrieben.

Wie unsere Rohrpostadresse enthält auch eine IP-Adresse nicht nur die Information, *welches* Gerät gemeint ist, sondern auch, *in welchem Netzwerk* sich dieses befindet. Wir können eine IP-Adresse deshalb gedanklich in zwei Teile zerlegen:

- Der **Netzwerkteil** bezeichnet das lokale Netzwerk.
- Der **Geräteteil** bezeichnet ein bestimmtes Gerät innerhalb dieses Netzwerks.

Nehmen wir als einfaches Beispiel die IP-Adresse 17.42.1.10 und nehmen vorerst an, dass die ersten 24 Bits den Netzwerkteil bilden. Dann können wir uns die Adresse schematisch so vorstellen:

```
17.42.1 | 10
Netzwerk  Gerät
```



den Netzwerkteil ausmachen.

Wenn Trudi von ihrem Computer (mit IP-Adresse 17.42.1.10) Daten an Reto (IP-Adresse 16.62.5.21) schicken möchte, so stellt sie fest, dass sich die Netzwerkteile unterscheiden. Reto befindet sich also in einem anderen Netzwerk und das Paket muss über Trudis Default Gateway weitergeleitet werden.

Möchte Trudi aber eine Nachricht an 17.42.1.14 schicken, so stimmen die Netzwerkteile überein. Die Nachricht kann innerhalb des gleichen lokalen Netzwerks zugestellt werden.



AUFGABE 9: NAH ODER FERN?

Peter behauptet: *Die Rechner 172.16.4.20/25 und 172.16.4.230/25 befinden sich im gleichen Netzwerk, weil die ersten drei Zahlen ihrer IP-Adressen gleich sind.* Hat er recht? Begründe deine Antwort.

Pakete in der Rohrpost

Bevor wir uns anschauen, wie ein IP-Paket seinen Weg durch mehrere Netzwerke findet, wechseln wir nochmals den Kontext und wenden uns den Pipe Islands zu.

Zur Erinnerung: Auch dort arbeitet die Verteilabteilung mit Paketen. Diese Pakete sind mit Rohrpostadressen adressiert. Trudis Tischlerei hat beispielsweise die Adresse 302 - 10; der Aussichtsturm auf derselben Insel die Adresse 302 - 14. An der Inselnummer erkennt Trudis Verteilabteilung sofort, dass beide Gebäude auf Tübingen liegen.

Nehmen wir also an, Trudi möchte eine Nachricht **innerhalb ihrer Insel** an den Aussichtsturm schicken. Sie übergibt die Empfängeradresse 302 - 14 zusammen mit dem Inhalt der Nachricht an die Verteilabteilung³.

Die Verteilabteilung verpackt die Nachricht in ein Paket und schreibt die eigene Rohrpostadresse 302 - 10 als Absender sowie 302 - 14 als Empfänger darauf. Anhand der Inselnummer stellt sie fest, dass sich das Ziel auf derselben Insel befindet. Das Paket muss also nicht über eine Verteilzentrale in ein anderes Inselnetzwerk geschickt werden.

Die Verteilabteilung besitzt allerdings kein eigenes Versandmedium und greift deshalb auf die darunterliegende Rohrpostabteilung zurück. Diese arbeitet mit Gebäudenamen und nicht mit Rohrpostadressen. Die Verteilabteilung muss deshalb herausfinden, welcher Gebäudename zur Rohrpostadresse 302 - 14 gehört.

Wenn diese Zuordnung noch nicht bekannt ist, kann sie eine Nachricht an alle Gebäude der Insel schicken und fragen, welches Gebäude die Rohrpostadresse 302 - 14 besitzt. Das entsprechende Gebäude meldet sich zurück. Wir nehmen an, es handelt sich um den

³Wir ignorieren hier vorerst die Transportabteilung, die hat aber natürlich auch noch ihre Finger im Spiel! Wir kommen im nächsten Kapitel darauf zurück.

Aussichtsturm Tübingen.

Die Verteilabteilung übergibt nun ihr Paket an die Rohrpostabteilung mit der Aufforderung, dieses an den *Aussichtsturm Tübingen* zu schicken. Das Paket wird in eine Rohrpostdose verpackt, die Dose mit dem Gebäudenamen adressiert und über die Rohrpost zugestellt. Beim Aussichtsturm angekommen, wird die Dose geöffnet und das darin enthaltene Paket an die Verteilabteilung weitergegeben. Diese erkennt ihre eigene Rohrpostadresse als Ziel und reicht die enthaltenen Daten an die nächsthöhere Schicht weiter.

Wie sieht die Situation aus, wenn Trudi eine Nachricht an Reto schicken möchte, der sich **auf einer anderen Insel** befindet? Reto hat die Rohrpostadresse 712 - 21. Schon anhand der Inselnummer erkennt Trudis Verteilabteilung, dass das Ziel nicht auf Tübingen liegt. Das Paket muss also die eigene Insel verlassen.

Dazu übergibt die Verteilabteilung das Paket an die Rohrpostabteilung mit der Aufforderung, es an die *Hauptverteilzentrale Tübingen* zu schicken. Diese Verteilzentrale ist für die Gebäude auf Tübingen quasi das Tor zu den übrigen Pipe Islands, also das Gegenstück zum Default Gateway in einem Computernetzwerk.

Die Rohrpostabteilung verpackt das Paket in eine Rohrpostdose und adressiert diese lokal an die Hauptverteilzentrale. Dort angekommen wird die Dose geöffnet und das Paket an die Verteilabteilung der Zentrale weitergereicht. Diese schaut sich die Rohrpostadresse des Empfängers an und entscheidet, an welche nächste Verteilzentrale das Paket weitergegeben werden muss.

Wie genau sie diese Entscheidung trifft, lassen wir für den Moment noch offen. Wichtig ist zunächst nur: Das Paket behält auf seiner Reise seine Rohrpostadressen von Absender und Empfänger. Für jeden einzelnen Abschnitt wird es aber in eine neue Rohrpostdose verpackt, deren Gebäudenamen lediglich den nächsten lokalen Transportabschnitt adressieren.

So wandert das Paket von Verteilzentrale zu Verteilzentrale, bis es Rohrbach erreicht. Dort erkennt die Verteilabteilung anhand der Inselnummer 712, dass sich das Ziel auf der eigenen Insel befindet. Nun wird das Paket wie im ersten, lokalen Fall an das Gebäude mit Hausnummer 21, also an Retos Restaurant, zugestellt.

Die Reise eines IP-Pakets

Zurück zur Realität: Wie finden Pakete auf dem Internet Layer ihren Weg von A nach B? Das Prinzip ist eigentlich analog zu dem auf den Pipe Islands, wir müssen einfach ein paar Wörter und Begriffe ersetzen. Die Rohrpostdose ist ein Ethernet-Frame, die Rohrpostabteilung ist der Link Layer, die Pakete sind IP-Pakete, die Verteilabteilung ist der Internet-Layer und Verteilzentralen nennen wir Router. Wenn wir dazu noch Rohrpostadressen durch IP-Adressen und Gebäudenamen durch MAC-Adressen ersetzen, kommen wir der Sache schon ziemlich nahe!

Nehmen wir an, ein Rechner A möchte Daten mit einem Rechner B austauschen. Dazu braucht A B's IP-Adresse. A baut also ein IP-Paket mit der gewünschten Nachricht und schreibt die

eigene IP-Adresse als Absender und B's IP-Adresse als Empfänger. Anhand seiner eigenen IP-Adresse und der Netzmaske kann Rechner A ebenfalls herausfinden, ob B sich im gleichen oder in einem entfernten Netzwerk befindet.

Im ersten Fall, also wenn beide Rechner im gleichen lokalen Netzwerk angeschlossen sind, kann A die Nachricht direkt an B schicken, ohne über einen Router zu gehen. Er muss nur anhand von B's IP-Adresse dessen MAC-Adresse bestimmen. Dies kann er mit einem so genannten ARP-Request tun.

ARP



ARP (Address Resolution Protocol) ist ein Protokoll, um für eine IP-Adresse die zugehörige MAC-Adresse zu bestimmen. Wie genau ein solcher ARP-Request aussieht ist für uns jetzt nicht im Detail relevant. Wichtig fürs Verständnis ist aber, dass ARP quasi den *Leim* zwischen den untersten beiden Layern darstellt: Es übersetzt Adressen des Internet Layers (IP-Adressen) zu Adressen des Link Layers (MAC-Adressen).

Nun, wo B's MAC-Adresse bekannt ist, kann das IP-Paket als Dateneinheit an den Link Layer übergeben werden, welcher dieses in ein Ethernet-Frame verpackt und an B's MAC-Adresse verschickt. Bei B angekommen entpackt der Link-Layer das Frame, entnimmt das IP-Paket und übergibt es dem Internet Layer, welcher die Nachricht auslesen kann.

VERKAPSELUNG (MATRJOSCHKA-PRINZIP)



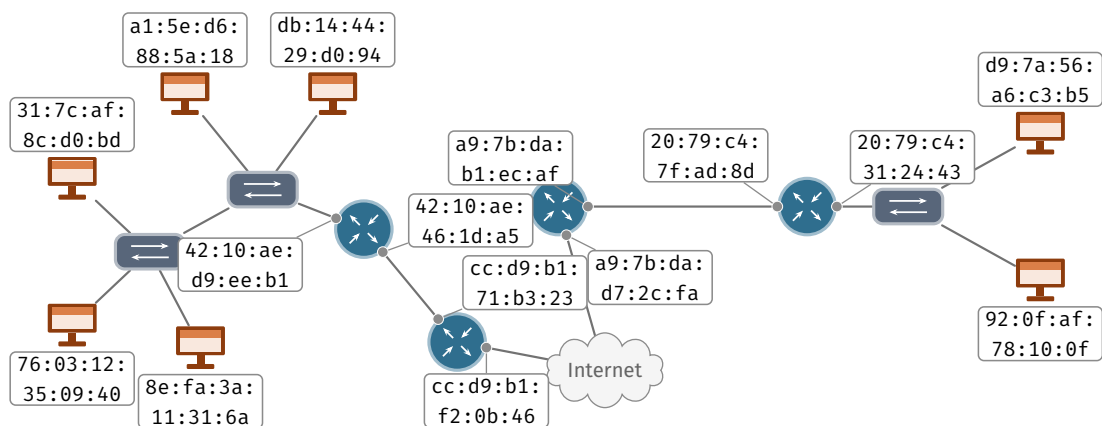
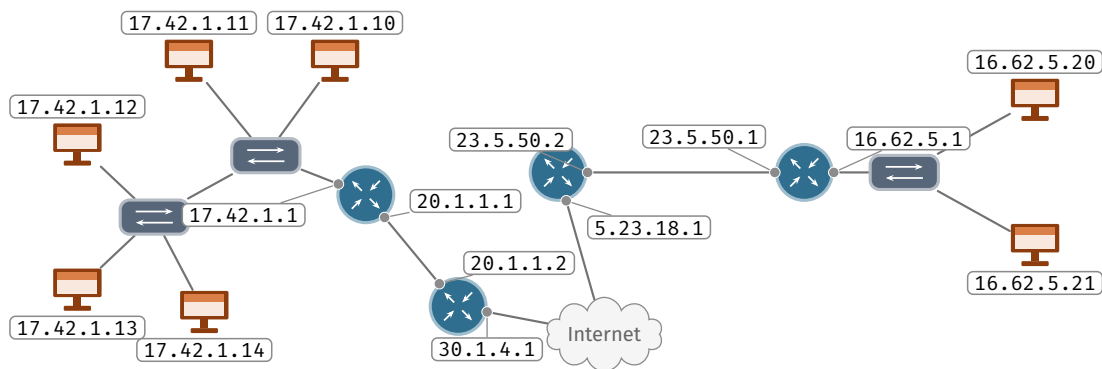
Hier begegnet uns nun auch endlich die Matroschka aus dem Titel dieses Kapitels. Die verschiedenen Schichten arbeiten nämlich ein wenig wie die bekannten russischen Holzpuppen, bei denen eine Puppe in einer grösseren steckt.

Die Internetschicht erstellt ein IP-Paket. Damit dieses Paket über ein lokales Ethernet-Netzwerk übertragen werden kann, übergibt sie es an den darunterliegenden Link Layer. Dieser behandelt das gesamte IP-Paket als Nutzdaten und verpackt es wiederum in ein Ethernet-Frame.

Eine Dateneinheit einer höheren Schicht steckt also jeweils als Nutzlast in der Dateneinheit der darunterliegenden Schicht – ganz ähnlich wie eine Matroschka in der nächsten.

Sind A und B in unterschiedlichen Netzwerken zu Hause, so schickt A das IP-Paket nicht direkt an B, sondern ebenfalls via einem Ethernet-Frame an den Default Gateway, den Router des Netzwerkes. Von dort aus wandert es dann Schritt für Schritt ins Zielnetzwerk, wo dann wieder der erste, lokale Fall abgespielt wird.

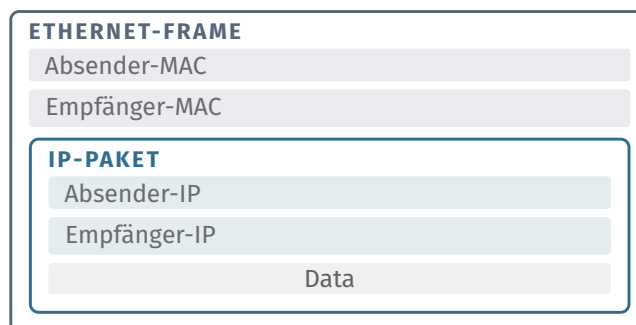
Am besten verfolgen wir die Reise eines IP-Pakets in einem Beispiel, Schritt für Schritt. Wir betrachten dazu wiederum unser Beispiel-Netzwerk. Im oberen Bild sind die IP-Adressen angegeben, im unteren die MAC-Adressen der Geräte.



BEISPIEL: EIN IP-PAKET AUF REISEN

Der Rechner mit IP-Adresse 17.42.1.10 möchte eine Nachricht an 16.62.5.21 schicken.

Wir verfolgen, welche Pakete bzw. Frames dazu in und zwischen den Netzwerken verschickt werden. Dies tun wir nach folgendem Schema:

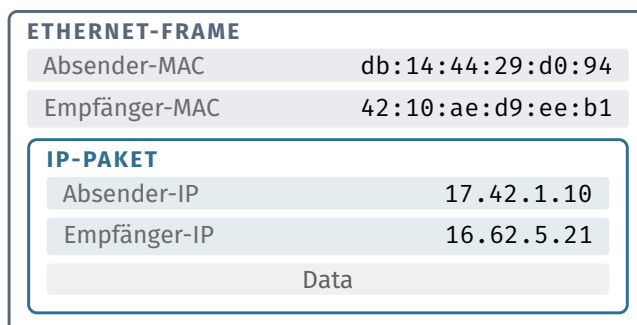


Das Schema repräsentiert ein IP-Paket, das in ein Ethernet-Frame verpackt ist. Wir betrachten hier jeweils nur die Absender- und Empfänger-Adressen, übrige Felder wie Länge oder Checksums interessieren uns hier nicht.

Zu Beginn findet der Absender heraus, dass die Empfänger-Adresse sich nicht im gleichen Netz befindet. Das IP-Paket wird also an den Default-Gateway, den Router mit



IP 17.42.1.1 geschickt. Dies geschieht, in dem das IP-Paket in ein Frame verpackt wird, welches an diesen Router, also seine MAC-Adresse adressiert ist:

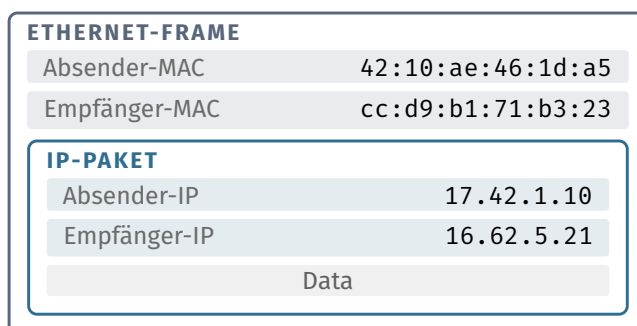


Der Router entpackt das Frame, betrachtet das IP-Paket, und leitet es weiter an den nächsten Router, und zwar in einem neuen Frame. Die Adressierung des IP-Pakets bleibt dabei unverändert!

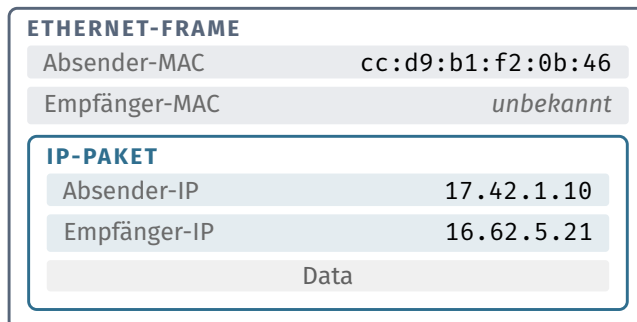
ERINNERUNG: MATRJO SCHKA



In unserer Matrjoschka-Analogie öffnet der Router also die äusserste Puppe: Er entfernt das Ethernet-Frame und erhält das darin enthaltene IP-Paket. Für den nächsten lokalen Netzwerkabschnitt wird dieses anschliessend wieder in eine neue äussere Puppe (ein neues Ethernet-Frame mit neuen MAC-Adressen) verpackt.

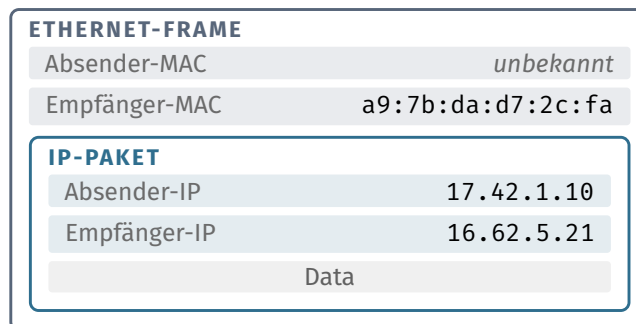


Der nächste Schritt führt in die "Cloud", deshalb wissen wir nicht genau, wohin das Frame adressiert ist:

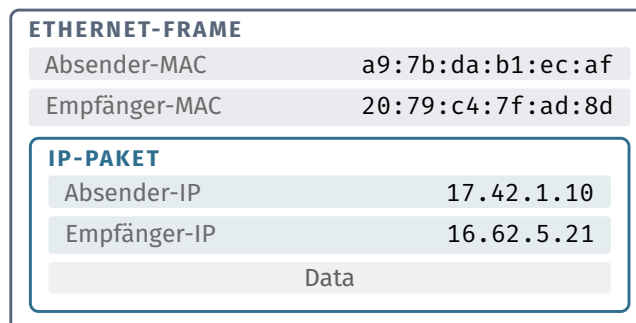




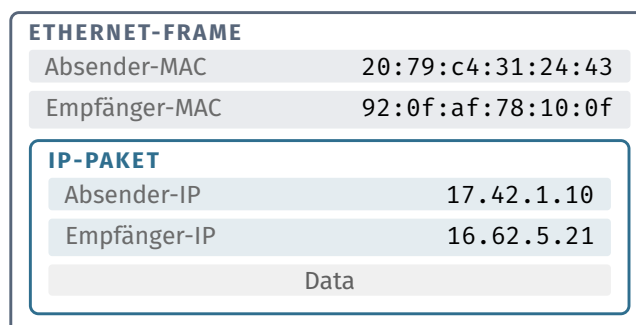
Und wenn das Frame aus der Cloud herauskommt, ist es an den Router mit MAC-Adresse a9:7b:da:d7:2c:fa adressiert:



Im nächsten Schritt erreichen wir das Zielnetzwerk:



Das Paket liegt jetzt beim Router mit IP-Adresse 16.62.5.1, welcher sich im gleichen Netzwerk wie das Zielgerät befindet. Das Paket kann also in einem Frame direkt dorthin geschickt werden:



Das Paket ist am Ziel angekommen! Der Rechner mit IP 16.62.5.21 betrachtet das Paket und stellt fest, dass es tatsächlich an sich adressiert ist, und kann die Daten an die oberen Layer weiterleiten.

Wie du feststellst, bleiben die Absender- und Empfänger-IP-Adressen auf dieser Reise gleich. Für jeden lokalen Transportabschnitt wird das IP-Paket lediglich in ein neues Ethernet-Frame mit passenden MAC-Adressen verpackt.