Netzwerk

Stable Diffusion Art “Networks”

Inhaltsverzeichnis

For now I 1:1 copied the structure of the pptx I have here. I will also mostly adapt the content, if you have Bedenken right away, here we go

1. Internet und Adressierung

1.1. Historisches

Das **Internet** entstand primär aus dem 1969 entwickelten ARPANET (Advanced Research Project Agency Networkt) Projekt an dem vier US-Amerikanische Universitäten beteiligt waren, deren Ziel die weitläufige Vernetzung von Rechnern zur gemeinsamen Nutzung von Ressourcen war. Die Idee größere Netzwerke aufzubauen stammte dabei schon aus den Anfängen der 1960er Jahre und viele Gruppen forschten und arbeiteten daran (z.B. auch das US Militär). Dieses Projekt war auch eines der ersten, die das **TCP/IP** Protokoll implementiert haben (RFC 9293 - der aktuelle Standard).

In der Folge wurden vor allem weitere Universitäten in Europa und den USA in Netzwerken miteinander verknüpft bis die Technologie so weit ausgereift war, dass es zu einer regelrechten Explosion kam, die dazu führte, dass die meisten technischen Geräte heute miteinander **physikalisch** verbunden sind.

Please feel free to add information you deem important at any point, but please be aware that we do only have so much time to discuss things

Das **World Wide Web** ist in den 80er Jahren am CERN (Conseil européen pour la recherche nucléaire) entwickelt werden und *nicht* synonym zum Internet zu verstehen. Spricht man vom Internet, dann ist primär die physikalische Verbindung, d.h. die Hardware der Netzwerke gemeint. Das WWW ist jedoch ein Dienst, d.h. eine Software. Die Grundlage bilden dabei aufeinander verweisende Hypertext-Dokumente, die auf verschiedensten Rechnern abgespeichert sind. Das WWW benutzt also das Internet als Medium, ist jedoch nur einer von vielen Diensten, die das tun. Weitere Beispiele für Dienste sind **E-Mails**, **FTP** (file transfer protocol), **HTTP** (Hyper-Text-Transfer-Protocol) oder das **XMPP** (extensible messaging and presence protocol - für Chat-Anwendungen).

1.2. Wiederholung Bits, Bytes und Hex

In der Informatik werden andere **Stellenwertsysteme** als in der schulüblichen Mathematik verwendet. Während dort das Zehnersystem Einzug gehalten hat ist für Computer das Zweiersystem geeignet (Strom an, Strom aus!).

Das bedeutet konkret: Statt in Zehnerpotenzen zu denken und daraus die Zahlen "zusammenzubauen" wie in der fünften Klasse (z.B. $12985 = 10000 + 2000 + 900 + 80 + 5$) werden **Binärzahlen** aus **Zweierpotenzen** zusammengebaut.

Das erscheint uns auf den ersten Blick merkwürdig - das liegt letztendlich aber nur an der Gewöhnung an das Zehnersystem.

Zweierpotenzen für die Umrechnung									
512	256	128	64	32	16	8	4	2	1

Beispiele

- Binär: 0111 1010 lässt sich dezimal schreiben als:

  $0 \cdot 128 + 1 \cdot 64 + 1 \cdot 32 + 1 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 122$

- Dezimal: 195 lässt sich umwandeln zu:

  $195 = 1 \cdot 128 + 1 \cdot 64 + 0 \cdot 32 + 0 \cdot 16 + 0 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1$

  und entspricht somit in Binärdarstellung als 1100 0011

DODO Expliziten Algo angeben, 0b und 0x Notation ansprechen

Beschränkt man sich auf Zahlen, die durch 8 Bit (also 1 Byte!) dargestellt werden, dann ist der Zahlenraum also auf die Zahlen von 0 bis 255 beschränkt, da 255 binär 1111 1111 entspricht.

Aufgabe

Wandeln Sie jeweils in Binär bzw. Dezimaldarstellung um.

- 253
- 1110 0111
- 187
- 1011
- 11 0101 1011

Hexadezimale Darstellung von Zahlen

Große 0-1 Folgen sind für Menschen schwer zu lesen. Zum Test - welcher Dezimalzahl entspricht die folgende Binärzahl?

$$01010001101011110000110011101000$$

Es ist natürlich 1370426600! Um solche $0-1$-Folgen lesbarer zu machen wird häufig die **hexadezimale** Darstellung von Zahlen verwendet, diese verwendet nicht das Zweiersystem, sondern das "Sechzehnersystem". Da wir mit vier Stelle einer Binärzahl gerade 16 Zahlen darstellen können, teilen wir also die obige Zahl in Viererblöcke ein:

$$0101\ 0001\ 1010\ 1111\ 0000\ 1100\ 1110\ 1000$$

Jeder dieser Viererblöcke entspricht jetzt einer "Ziffer" im Sechzehnersystem. Da es zu Uneindeutigkeiten kommen würde, wenn man "10" als Ziffer verwendet (Dieses Problem hatte man beim Zweiersystem natürlich nicht) müssen andere Zeichen als "Ziffern" verwendet werden. Man hat sich hier auf die Zeichen a bis f geeinigt, es gilt also:

$(0000)_2 = (0)_{16}$	$(0001)_2 = (1)_{16}$	$(0010)_2 = (2)_{16}$	$(0011)_2 = (3)_{16}$
$(0100)_2 = (4)_{16}$	$(0101)_2 = (5)_{16}$	$(0110)_2 = (6)_{16}$	$(0111)_2 = (7)_{16}$
$(1000)_2 = (8)_{16}$	$(1001)_2 = (9)_{16}$	$(1010)_2 = (a)_{16}$	$(1011)_2 = (b)_{16}$
$(1100)_2 = (c)_{16}$	$(1101)_2 = (d)_{16}$	$(1110)_2 = (e)_{16}$	$(1111)_2 = (f)_{16}$

Die Klammern zeigen dabei an, wie die Zeichen innerhalb interpretiert werden sollen (das hätte man auch schon bei der Umwandlung vom Binär- ins Dezimalsystem machen können, dort ist die Mehrdeutigkeit aber nicht so dramatisch.)

Die obige Binärzahl enspricht also der folgenden Hex-Zahl:

$$5\ 1\ a\ f\ 0\ c\ e\ 8$$

Häufig gruppiert man diese noch zu Zweierpaketen, dann entspricht eine dieser Hex-Gruppen genau einem Byte ($2 \cdot 4$ Bit!)

$$51\ af\ 0c\ e8$$

Nochmal veranschaulicht:

$$51\ af\ 0c\ e8$$

0101 0001 1010 1111 0000 1100 1110 1000

Aufgabe

Wandeln Sie jeweils von Binär- in Hexadezimaldarstellung oder umgekehrt um:
- 0110 0111
- $ff\ af$
- 1010 1100 0001 1001
- $0e\ c1$ 10

1.3. IP-Adressen

IP(v4) ist die erste Version des Internet Protocols, das weltweit verbreitet und eingesetzt wurde. Eine solche IP-Adresse besteht aus einer 32-stelligen Binärzahl, die jeweils einem Netzwerk-Interface im Internet zugeordnet ist und so eine Identifizierung möglich macht. Netwerken voin z.B. Providern wird dann ein ganzer Block an Adressen zugeordnet (siehe unten).

Zur besseren Lesbarkeit wird die Adresse üblicherweise in vier 8-stellige Böclke aufgeteilt und (für den Menschen) dann in Dezimalschreibweise abgebildet werden - jeweils durch einen Punkt getrennt.

Beispiel

01101100001000010000111100110111

01101100 00100001 00001111 00110111

108.33.15.55

Kurze Denkaufgabe!

Wie viele solcher IP-Adressen können gebildet werden?

Es sind natürlich 2^{32}, also etwa $4,3$ Millarden Adressen.

Aufgabe

Wandeln Sie jeweils von Binär- in Dezimaldarstellung oder umgekehrt um:
- 88.215.213.26
- 0101 1000 0110 0011 0011 1111 1001 0011

IP(v6) Da die Anzahl der möglichen IPv4-Adressen doch recht klein ist im Vergleich zu der Anzahl an Geräten und Netwerken wurde ein neuer Standard entwickelt. Die Adressen in diesem Standard sind 128 Bit lang, somit gibt es 2^{128} unterschiedliche IPv6-Adressen.

Hier kommt auch die hexadezimale Darstellung ins Spiel, denn die Größe der Zahlen macht es sonst wieder unlesbar. Die Adresse wird also in 8 Blöcke zu je 16 Bit eingeteilt, die wiederum hexadezimal kodiert werden, a.so z.B.:

$$2600 : 1f18 : 001f : db00 : 807b : f1f4 : d01b : 30b1$$

Hinweis

Ob dieses Format wirklich lesbarer ist oder wirklich sein muss sei einmal dahingestellt..

Die Tatsache, dass IPv6 Adressen noch nicht “der Standard” ist liegt an zweierlei:

1. Es gab andere technische “Tricks”, mit denen die Limitierung von IPv4 abgefedert wurde.

2. Es braucht Zeit, bis sich Dinge durchsetzen, die neu sind. (Es sei nochmal darauf hingewiesen, dass in vielen Betrieben Java 8 der Standard ist z.B.)

1.4. Adressvergabe

IP-Adressblöcke werden an Unternehmen und Organisationen vergegeben, die diese ihrerseits wieder verteilen können.

Vergebene Adressblöcke: Findet man z.B. hier

Die Einträge sind dabei folgendermaßen zu lesen: $\mathbf{84.39.}\frac{\mathbf{64.0}}{\mathbf{19}}$

Die Zahl 19 bedeutet hier: Für alle Adressen aus dem Adressblock gilt, dass die ersten 19 Bit fest vorgegeben sind und die restlichen Stellen beliebig verwendet werden können, d.h. die rot markierten Stellen sind fix:

$$84.39.64.0 = \text{xxxx xxxx . xxxx xxxx . xxx}x \text{ xxxx. xxxx xxxx}$$

Kurze Denkaufgabe

Wie viele Adressen umfasst damit der Adressblock aus dem Beispiel, d.h. wie viele Adressen kann der Kunde hier nutzen

Da 13 Bit für den Kunden verfügbar sind, kann er 2^{13} Adressen verwenden, also alle Adressen von 94.39.64.0 bis 84.39.95.255

In der RFC 1918 werden bestimmte Adressblöcke für private Netze festgelt.

> **Zitat**
>
> The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the IP address space for private internets:
> 1. 10.0.0.0 – 10.255.255.255 (10/8 prefix)
> 2. 172.16.0.0 – 172.31.255.255 (172.16/12 prefix)
> 3. 192.168.0.0 – 192.168.255.255 (192.167/16 prefix)
>
> We will refer to the first block as “24-bit block”, the second as “20-bit block” and to the third as “16-bit block”. Note that (in pre-CIDR notation) the first block is nothing but a single class A network number, while the second block is a set of 16 contiguous class B network numbers and third block is a set of 256 contiguous class C network numbers.
>
> An enterpreise that decides to use IP addresses out of the adress space defined in this document can do so without any coordination with IANA or an Internet registry. The address space can thus beused by many enterprises. Adresses within this private address space will only be unique within the enterprise, or the set of enterprises which choose to cooperate over this space so they may communicate with each other in their own private internet.

Kurz zusammengefasst: die oben erwähnten Bereiche werden **nicht für öffentliche Adressen im Internet vergeben**. Zur öffentlichen Kommunikation wird dem **gesamten Teilnetz** vom Internetprovider eine IP-Adresse aus einem seiner Adressblöcke zugeteilt. Jedes Gerät des Teilnetztes ist nach außen hin dann nur unter dieser Adresse sichtbar.

Dynamische Adressierung: In beiden Fällen (lokal/öffentlich) gibt es sowohl die Möglichkeit, dass man immer die gleiche, also eine feste Adresse hat, als auch die Möglichkeit, dass in gewissen Absständen bzw. bei jedem Beitritt zum Netz eine beliebige verfügbare Adresse zugeteilt wird, also **dynamisch**.

1.5. URL und DNS

Eine **URL** (uniform resource locator) ist eine anwenderfreulichere Benennung von Internetressourcen. In der Regel werden Websiten bzw. deren Ordnerstruktur so benannt.

Beispiel: https://#text(red)[www.wgg-neumarkt.de]

DNS (domain name system):

Das oben rot gedruckte ist die Domain der Website. Ein DNS-Server übersetzt jede Domain, die beispielsweise ein Anweder in die Adresszeile seines Browsers eingibt, in die dazu gehörige IP-Adresse, damit anschließend die Verbindung zu dem Rechner, der diese Domain hostet, aufgebaut werden kann.

Aufgabe

Finden Sie die zugehörige IP-Adresse zu www.wgg-neumarkt.de und zu www.google.de. Recherchieren Sie dazu geeignete tools im Internet.

1.6. Routing

Aufgrund der Struktur des Internets verlaufen Wege nicht "direkt", sondern die Kommunikation kann über verschiedenste Geräte verlaufen. Möchte man die **Route** nachverfolgen, kann unter Windows beispielsweise der *tracert* Befehl verwendet werden. Im folgenden beispiele wurde

```
tracert wgg-neumarkt.de
```

ausgeführt:

```
Routenverfolgung zu wgg-neumarkt.de [85.236.38.213]
über maximal 30 Hops:

  1    <1 ms    <1 ms    <1 ms  fritz.box [192.168.178.1]
  2    58 ms    15 ms     9 ms  p3e9bf3ff.dip0.t-ipconnect.de [62.155.243.255]
  3    11 ms    11 ms    10 ms  m-ec4-i.M.DE.NET.DTAG.DE [217.5.69.162]
  4    12 ms    30 ms    11 ms  87.190.232.83
  5    11 ms    11 ms    11 ms  c01-vl901.muc01.net.internetx.com [85.236.32.138]
  6    12 ms    12 ms    11 ms  web.wgg-neumarkt.de [85.236.38.213]

Ablaufverfolgung beendet.
```

Der erste "Hop" ist dabei die hauseigene Fritzbox, die die Verbindung nach außen regelt. Es folgen einige Hops über Zwischenstationen, bis offenbar der Adressbereich der WGG-Homepage erreicht wird (85.236.32.138). Von dort aus ist es dann nur noch ein Sprung bis zur eigentlichen Website.

DODO: Einfügen Bild von Routing aus dem Schulnetz -> mehrere Hops im lokalen Netzwerk

Im Allgemeinen sind die Strukturen und technischen Details noch wesentlich komplexer. Für uns reicht aber dieser grobe Überblick.

So.. there is surely a lot more to talk about, you mentioned CIDR notation, that I stuff in above at IPv4 yus? So you really want me to torment them with subnetting and such as well? What else shall be added?

1.7. Organisationen

Es gibt mehrere Organisationen, die mit der **"Verwaltuing"** des Internets beschätigt sind:

1. ICANN: Internet Corporation for Assigned Names and Numbers
   Diese Organisation koordiniert die Vergabe von einmaligen Namen und Adressen im Internet. Dazu gehören insbesondere die Koordination des Domain Name Systems und die "IANA-Funktion" (Internet Assigned Names and Numbers).
2. Réseaux IP Européens Network Coordination Centre
   Eine ähnliche Organisation, die aber speziell für Europa, den Nahen osten und Teile von Zentralasien zuständing ist.
3. IETF: Internet Engineering Task Force
   Hier wird im Wesentlichen die technische Weiterentwicklung des Internets vorangetrieben.
4. RFC Request for Comments
   In diesem Skript bereits mehrfach erwähnt wurden RFC Memos, die gewisse Standards vorgeben. Diese Standards müssen implementiert bzw. berücksichtigt werden, wenn neue Technologie in den Verbund des Internets aufgenommen werden wollen.

2. Topologie von Rechnernetzen

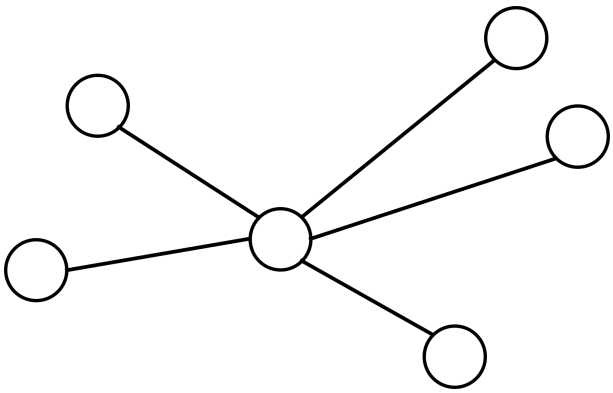
Geräte können auf verschiedene Arten miteinander verbunden werden, die jeweils verschiedene Vor- und Nachteile haben. Man spricht bei einer "Verbundart" auch von einer **Topologie**.

Im Folgenden werden die einzelnen Topologien jeweils nach den folgenden Kritierien untersucht bzw. bewertet:

1. **Störanfälligkeit**: Wie wirkt sich der Ausfall eines Rechners oder einer Verbindung aus.
2. **Systempflege**: Wie groß ist der Aufwand bei der Fehlersuche oder beim Erweitern.
3. **Kosten**: Wie hoch sind die Kosten für Leitungen oder spezielle Komponenten vergleichsweise.
4. **Übertragungsrate**: hierunter fällt auch die Frage nach möglichen Kollissionen.

Stern-Topologie

In der "realen Welt" ist diese Topologie die vorherrschende. Der Name ist im Wesentlichen selbsterklärend, es gibt es einen zentralen Rechner, der in der Mitte sitzt, die anderen werden "sternförmig" darum angeordnet. Das ist natürlich nur eine idealisierte netzwerktechnische Darstellung und stellt nicht notwendigerweise die tatächliche räumliche Positionierung dar.

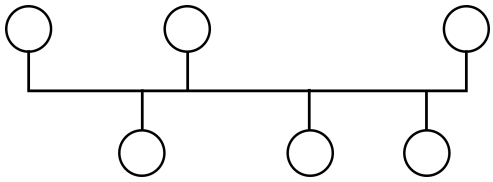


Analyse:

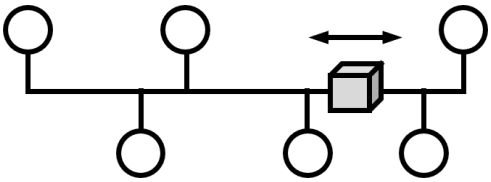
1. Der Ausfall des Zentralrechners ist problematisch, der Ausfall aller anderen Rechner dagegen nicht relevant.
2. Sowohl die Fehlersuche als auch die Erweiterung sind problemlos möglich.
3. Der Zentralrechner muss hochwertig sein.
4. Die Übertragungsrate kann, je nach Zentralrechner und Anzahl der Geräte insgesamt sein. Kollisionen können durch eine Steuerung des Zentralrechners aber vermieden werden.

Bus-Topologie

Hier sind alle Geräte mit einer zentral verlaufenden Leitung (Bus) verbunden, es gibt in diesem Sinne keine “zentrale” Stelle.



Bus - Topologie

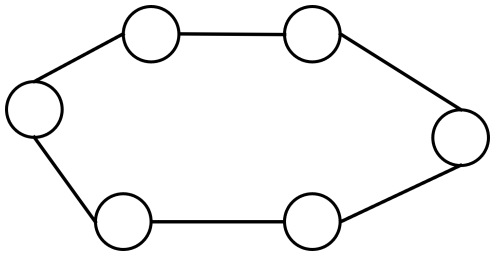


Token -Bus Topologie

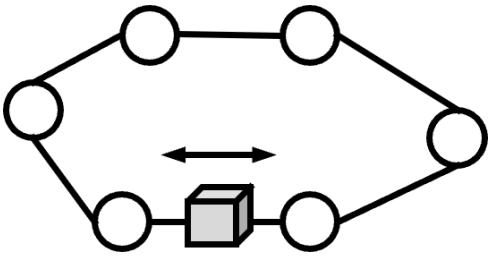
1. Der Ausfall des Bus ist problematisch, der Ausfall von Rechnern dagegen nicht.
2. Sowohl die Fehlersuche als auch die Erweiterung sind problemlos möglich.
3. Der Bus muss eine hohe Bandbreite besitzen, ansonsten gibt es keine teuren Komponenten.
4. Die Übertragungsrate hängt (quasi) alleine vom Bus ab. Bei hoher Last lassen sich Kollissionen nicht direkt vermeiden. Es kann der Ansatz eines sogenannten **Token Bus** verwendet werden.

Ring-Topologie

Auch hier ist der Name wieder selbsterklärend.



Ring - Topologie



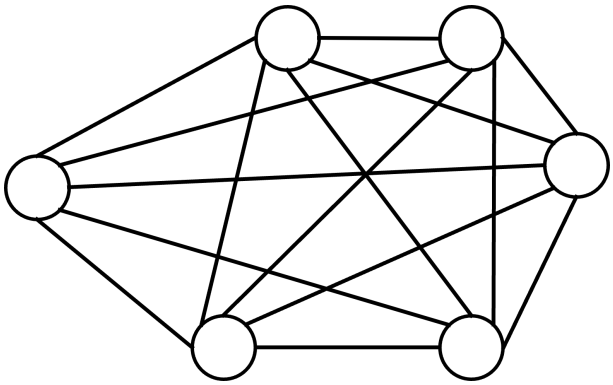
Token - Ring Topologie

I know that it is probably not important today, but I did not really understand the difference in da Token and non token setup. Do we just use fixed transmission rates with da second one here so there is always a Token that we get or not get?

1. Der Ausfall jedes Geräts unterbricht den Ring, führt aber nicht zu einem Totalausfall des Netzes.
2. Der Aufwand bei einer Fehler ist hoch, die Erweiterung geringfügig umständlicher als bei den vorherigen Topologien.
3. Es gibt keine besonders teuren Komponenten.
4. Die Übertragungsrate hängt von der Anzahl der Rechner ab. Kollisionen können analog zum Bus bei hoher Last auftreten und wiederum durch Verwendung eines Token-Rings vermieden werden.

Direkte Verbindungen

Bei dieser Bauart ist jeder Rechner mit jedem anderen verbunden.



1. Der Ausfall jedes Geräts ist unproblematisch.
2. Der Aufwand bei der Fehlersuche ist gering, das erweitern dagegen sehr aufwändig.
3. Keine teuren Komponenten, aber sehr viele Leitungen nötig $(\sim n^2)$ mit n der Anzahl der Rechner.
4. Die Übertragungsrate ist sehr hoch, Kollisionen sind dagegen selten.

Kombinationen

Natürlich sind auch Kombinationen verschiedener Arten von Rechnernetzen möglich:

DODO Bild

3. Protokolle

3.1. Grundlagen

Damit Kommunikation sinnvoll ablaufen kann müssen die "Spielregeln" der Kommunikation eindeutig festgelegt werden. Die Vereinbarungen für die technische Kommunikation sind in **Protokollen** festgehalten, die wiederum in den RFCs (siehe oben) spezifiziert werden.

Bevor wir uns detaillierter mit einzelnen Protokollen beschäftigen hier ein exemplarischer Ablauf einer Client-Server-Sitzung:

Server:
- Server starten (Port festlegen)
- Anmeldeversuch eines Clients: Akzeptanz oder Ablehnung
- Botschaft des Clients empfangen und entsprechend eines Algorithmus antworten.
- Ist eine Abschlussbedingung erreicht: Verbindung trennen
- ggf. Server beenden

Client:
- Anmeldung beim Server (IP-Adresse und Port notwendig)
- Antworten des Servers lesen und entsprechende Eingaben tätigen.
- ggf. Abschluss auslösen

The colleague had this example before he dives deeper, do you tink it is wise to do some simple Java Client Server Impl here? Me kinda wants to do da Polly, but me is unsure if that really is halpful cause you just use a bunch of Java intern stuff but then again.. you can start the server and the client and make them communicate.. hm

Port

Da auf jedem Rechner **viele Prozesse** ablaufen, die gegebenenfalls aber alle Datenpakete aus dem Netzwerk beziehen wollen, muss ein Unterscheidungsmerkmal getroffen werden. Dieses Merkmal wird **Port** genannt und entsprechend dieser **Portnummer** werden einkommende Datenpakte eindeutig zugeordnet werden. Die Portnummer ist eine Folge aus 16 Bits.

Beispiele für pbliche Portnummern
- http: hypertext transfer protocol: 80
- https: hypertext transfer protocol secure: 443
- FTP: file transfer protocol: 21
- POP3: post office protocol 3: 110
- SMTP: Simple mail transfer protocol: 25

Zu Datenpaketen

Die Daten, die während eines Kommunikationsprozesses übermittelt werden sollen, bzw. die Daten eines stetigen Datenstroms (Streaming) werden in viele kleine Portionen aufgeteilt, die individuell verschickt werden. (Daraus können sich natürlich Probleme ergeben, z.B.: was passiert, wenn einzelne Pakete verloren gehen?).

Neben den eigentlichen Daten, die Übertragen werden sollen, werden in der Regel Zusatzinformationen mit übertragen, z.B.:
- IP-Adressen von Sender und Empfänger
- MAC-Adresse von Sender und Empfänger: DODO: wo Mac-Adress-Erklärung einpflegen?
- ggf. fortlaufende Nummern der einzelnen Teil-Datenpakete einer Sendung
- Checksummen
- Timestamps
- Infos über die Größe des Pakets oder die Größe des Headers.

3.2. TCP/IP

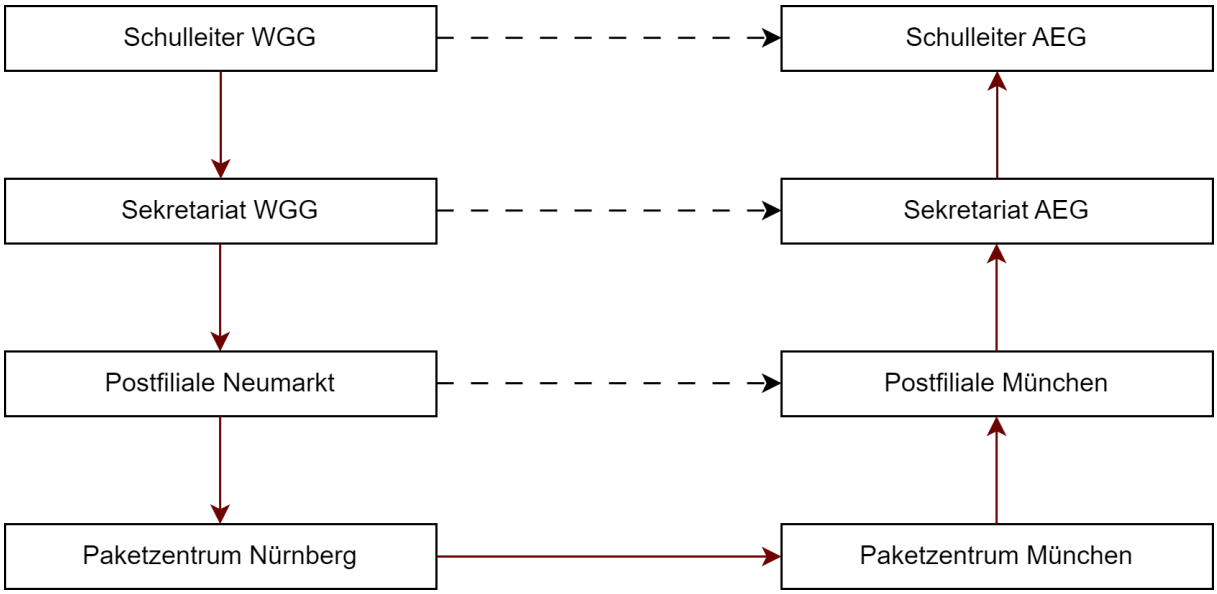
3.3. Ethernet

Those two are only mentioned here briefly with their RFCs, I would say those are good to look at a bit more? Or would you rather e.g. look at http? In general this whole section feels very fuzzy to me, I am not content with how da colleague did this, cause I did not feel I learned something maself^^

4. Das Schichtenmodell

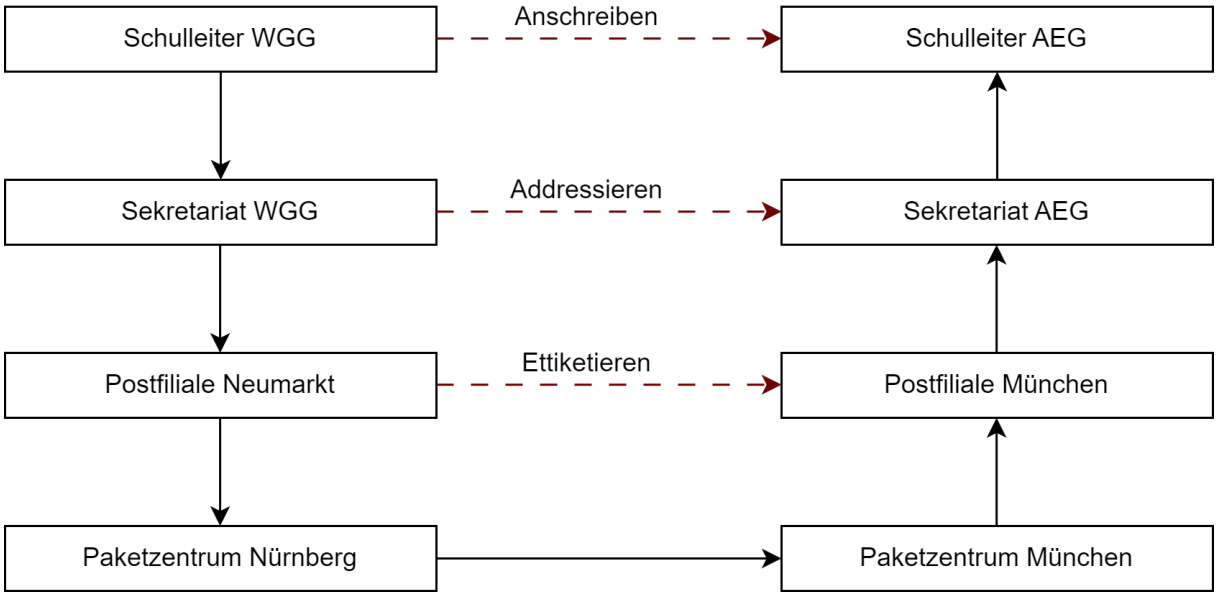
Die Übertragung von Daten kann sich auf mehreren Ebenen abspielen, die häufig sogar scharf voneinander getrennt werden können. Zur Beschreibung dieser einzelnzen Ebenen wird in der Netzwerktechnik das **OSI-Modell** verwendet (Open System Interconnection model). Im Abitur ist es üblich, dass nicht direkt ein technisches Beispiel bearbeitet wird, sondern ein Analogon. Deswegen soll auch hier mit einer Analogie begonnen werden:

Das folgende Beispiel stellt den Versand eines Pakets vom Schulleiter unserer Schulleiter zu einem anderen Schulleiter dar:



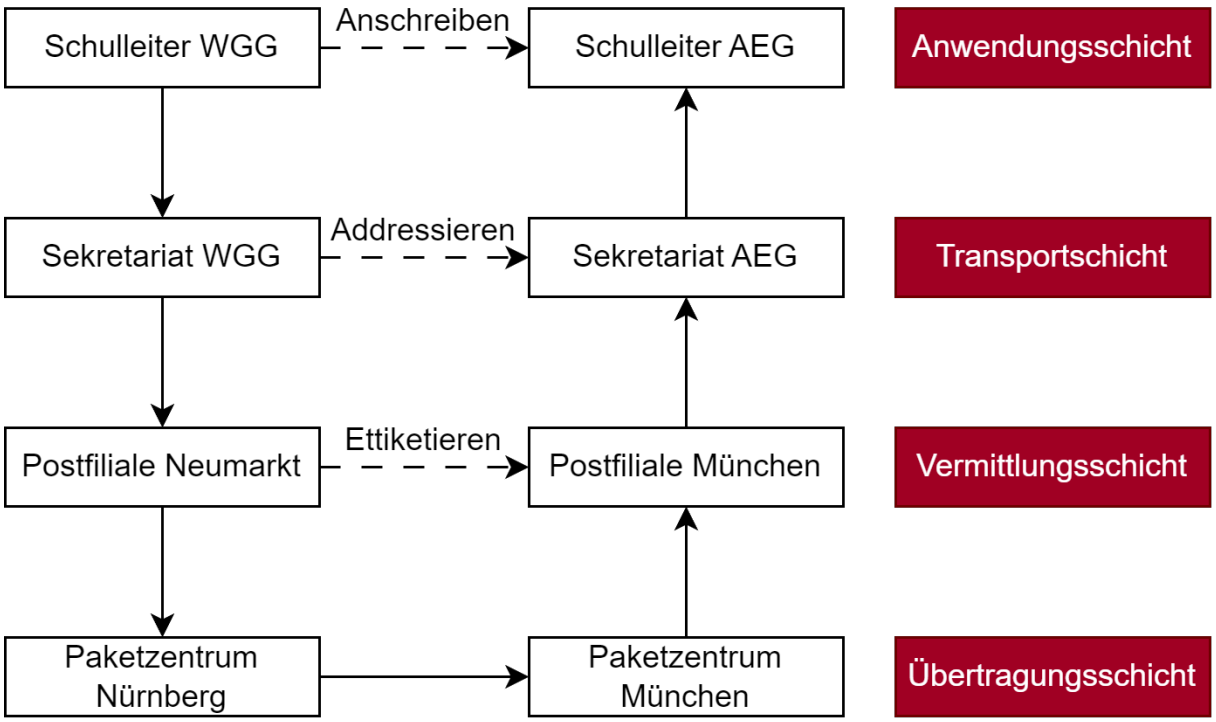
Die durchgezogenen Pfeile zeigen dabei den physikalischen Datenfluss (hier rot), d.h. in diesem Beispiel das Paket, das durch die verschiedenen Stationen wandert.

Für unser Schichtemodell besonders interessant sind aber die weiteren Ebenen. Denn auch hier findet Kommunikation statt, die indirekt in den physischen Daten mit eingebettet sind. Das könnte beispielsweise so aussehen:



1. Der Schulleiter wird in aller Regel ein persönliches Anschreiben beigelegt haben, das von seinem Kollegen gelesen wird.
2. Das Sekretariat des WGG muss das Paket addressieren, damit es einerseits von der Post korrekt zugestellt werden kann, andererseits gehen Pakete an Schulen zunächst im Sekretariat ein, d.h. die Weiterleitung an den Schulleiter geschieht durch das Sekretariat des AEG.
3. Die Postfiliale Neumarkt muss das Paket korrekt ettiketieren, damit die Weiterleitung nach München korrekt funktioniert.

Man kann jetzt eine (gewisse) Analogie zwischen diesem Schichtenmodell beim Postversand und der Kommunikation zwischen zwei Computern ziehen. Dies führt uns auf die Namen einiger Schichten des OSI-Modells:



Die **Übertragunsschicht** ist für den eigentlichen physikalischen Transport (zumindest in diesem Beispiel den Hauptteil des Weges) verantwortlich.

Die **Vermittlungsschicht** ist für die Weitervermittlung und die Wegewahl verantwortlich (Routing)

Die **Transportschicht** ist für das zuverlässige Senden und Empfangen, sowie für die Vollständigkeit der Daten zuständig.

Die **Anwendungsschicht** ist dann das eigentliche Ziel, d.h. hier der Informationsaustausch zwischen den Schulleitern.

5. Nebenläufige Prozesse

6. Das klassische Erzeuger - Verbraucher - Problem

7. Verklemmungen