Modul: Grundlagen digitaler Medientechnik

Die Technik des Internet

Martin Warnke

1. Das Internet Protocol

Alle Verhandlungen zur Gestalt des Internet, zu seiner Funktionsweise, insbesondere zu den Protokollen, die die technischen Konventionen dokumentieren sind, in den Requests For Comments, RFC, niedergelegt. Sie erinnern sich? Die schüchternen Jungs, die noch immer jemanden vermissten, der eigentlich das Sagen bei alledem hätte haben sollen, als sie schon selbst dabei waren, das Internet zu erfinden. Sie hatten vorgeschlagen, Kommentare zu Einfällen zu sammeln, auszugleichen, bis zum Konsens weiter zu entwickeln. In diesen RFC sind auch die großen Weichenstellungen des Internet dokumentiert, und man kann sie alle bequem über das Internet einsehen.

Die Internet Engineering Task Force IETF hat eine Sammlung davon angelegt, und unter der Nummer 791 wird zum Datum September 1981 das **Internet Protocol** beschrieben (http://tools.ietf.org/html/rfc791#ref-1). Das Internet Protocol beschreibt, wie Datenpakete von einem Computer eines lokalen Netzwerkes über das Internet zu einem anderen reisen, welche Informationen dazu vorhanden und gesammelt werden müssen. Von Erkenntniswert über diese technischen Details hinaus mag noch sein, wie ganz konkret die funktionale Schichtung in Protokolle abläuft, woran man ablesen kann, dass irgendwo die eine Schicht endet und eine andere beginnt.

Neben den RFC selbst, den Gründungsdokumenten des Internet, die allen ans Herz gelegt seien, die genau nachschlagen wollen (zu erreichen etwa unter http://www.ietf.org/rfc.html) stütze ich mich in diesem technischen Kapitel auf das Buch von Kurose und Ross zu Computernetzwerken, das sehr ausführlich und entsprechend dickleibig so ziemlich alles verhandelt, was man an Netzwerktechnik wissen wollen könnte:

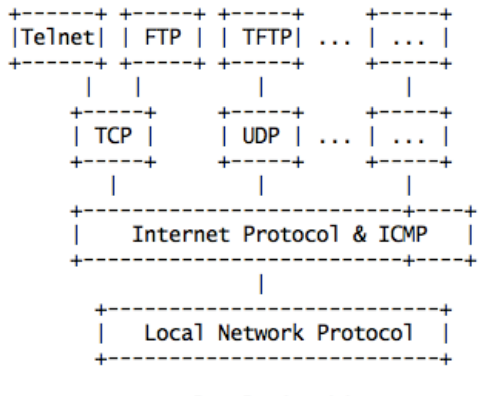
(Kurose, James F.; Ross, Keith W.: Computernetzwerke. München: Addison-Wesley, 2008.)

Aufgabe und Regelungsgegenstand des IP ist die Übertragung von Datenpaketen, die Datagramme heißen, von Absendern zu Empfängern, die jeweils mit einer Adresse fester Länge, der IP-Adresse, versehen sind. Notfalls regelt IP auch noch die Zerlegung dieser Pakete in kleinere Fragmente und deren späterem Wiederaufbau, denn man weiß ja nicht, ob eventuell zu durchquerende Teilstücke mit nur kleineren Datenhäppchen zurecht kommen können. Es wird bei dieser Protokollbeschreibung und selbst auch bei der Sicherung der Daten nur vom Header die Rede sein, vom Umschlag, in den die eigentliche Nutzlast verpackt ist.

Worum es sich im Inneren, um die Daten selbst, handelt, ist IP egal. Das ist die Netzwerkschicht-Grenze nach oben, zu dem, was in den Schriften als „user“ bezeichnet wird, den Programmen, die irgendetwas mit den Daten anfangen können und wollen. Oberhalb von IP kommen irgendwann die Anwendungen, E-Mail, das Web etwa, und die Daten-Nutzlast eines IP-Pakets bezieht sich auf diese Anwendungen. Aber dem Internet-Protokoll IP ist es gleichgültig, worum es sich dabei handelt. Das ist auch gut so, sonst müsste IP umgeschrieben werden, wenn radikal Neues mit den Daten in den Schichten darüber geschehen soll, wie es ja seit dem Bestehen des Internet ohne Unterlass sich ereignet.

Ebenso wird in IP darüber geschwiegen, wie die Technik der Teilnetze funktioniert, was also unterhalb von IP geschieht, sei es Ethernet, oder Funk oder irgendetwas anderes. Diese Gleichgültigkeit gegenüber der jeweiligen Netzwerkübertragungs-Technik markiert die Protokollschicht-Grenze nach unten.

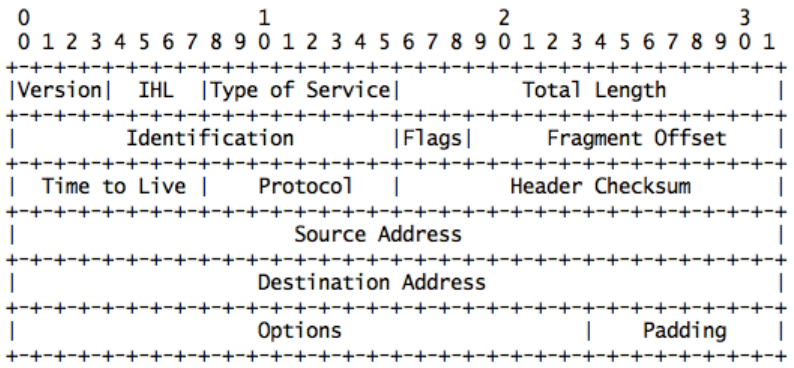
In schönster ASCII-Art stellt dann das RFC 791 dar, dass das IP unter TCP sitzt (das wird ein wenig später thematisiert), als eines von denkbaren Protokollen, dass sich darum kümmert, dass eine wirklich brauchbare Verbindung zwischen Computern zu Stande kommt, die auch diese Bezeichnung verdient hätte. Darunter ist das Protokoll irgendeines lokalen Netzes. „ICMP“ tritt in Aktion, wenn IP scheitert und Fehlermeldungen erzeugt werden müssen.



Protocol Relationships

(J. Postel, RFC 791, S. 5)

Die erste Funktion von IP ist die Adressierung der Datenpakete. Alles dafür Nötige steht im Header des Datenpakets, dem Vorspann, dem dann die eigentlichen Daten als Nutzlast folgen:



Example Internet Datagram Header

(RFC 791, S. 11)

Version gibt an, um welche Version des IP es sich im Folgenden handelt. Fast ausschließlich ist dies heutzutage die Version 4. Einige Hinweise zu Version 6 folgen später. Es steht in diesen vier Bits also die binäre vier: 0100. IHL sagt aus, wie lang der Header insgesamt ist, gezählt in Vielfachen von 32 Bit. Danach fangen die Daten an. Mit Type of Service haben die User, die Programme, die Datenpakete versenden, zum Ausdruck zu bringen, welche Präferenzen sie haben. Es gibt hier die Wahl zwischen niedrigen oder hohen Verzögerungszeiten (hoffentlich niedrig bei Internet-Telefonie), hohem oder niedrigem Datendurchsatz (bei großen Datenmengen, die Zeitnah am Empfänger sein sollen) oder dem Grad der Sicherheit. Besonders wichtige Pakete, etwa solche zur Kontrolle des Netzes selbst, bekommen überall eine Markierung, Routine-Daten, bei denen es nicht pressiert, überall keines. Ob allerdings ein Netzwerk-Router sich daran hält, das bleibt dann nur zu hoffen, es handelt sich also um eine Art Wunschzettel. Total Length gibt an, wie lang das Paket inklusive Datennutzlast ist. Die Länge dieses Feldes erlaubt Paketgrößen bis 64 kByte, meist sind Pakete aber viel kleiner, so ein bis zwei kByte (Kurose/Ross, S. 373). Die drei nächsten Felder, Identification, Flags und Fragment Offset braucht man, um Pakete im Verlauf der Übertragung noch weiter zu zerlegen, weil sie für ein Teilnetz-Abschnitt zu groß sind, und sie wieder richtig zusammenzusetzen. Die Fragmente bekommen dann eine Nummer, die Identification, eine Marke, ob sie das letze Bruchstück sind, und die laufende Nummer für‘s Zusammensetzen des Puzzles.

Time to Live ist ein interessantes Feld. Es ermöglicht nämlich einen Selbstzertörungs-Mechanismus für den Fall, dass ein Paket auf seinem Weg irgendwo in einer Schleife hängenbleibt und dann, gemeinsam

mit allen anderen verwaisten herumirrenden Paketen, das Internet verstopft. Dieses Feld ist maximal acht Bit breit. Die Zahl, die dort steht, wird um jeweils Eins heruntergezählt, wenn das Paket weitergereicht wird, erreicht diese den Wert Null, dann soll es verworfen werden. Spätestens jede Sekunde soll TTL heruntergezählt werden, was dann eine maximale Lebensdauer von 256 Sekunden, die man mit acht Bit zählen kann, oder rund viereinhalb Minuten ausmacht, meist aber viel weniger, weil ein Hop typischer Weise weniger als eine Sekunde braucht.

An diesem kleinen Feld mit überschaubarer Funktion soll klar werden, welche die Form einer Protokoll-Spezifikation wie des RFC 791 ist. Sie beschreibt nämlich in Englisch, was Sinn und Zweck und Funktion des Feldes sein sollen und wie die anderen Instanzen, die Software, die dieses Protokoll implementieren, funktionieren müssen. Wie das zu geschehen hat, steht hier nicht. Hier der kurze Abschnitt zum TTL, wie er im RFC 791 vorkommt: "Time to Live: 8 bits. This field indicates the maximum time the datagram is allowed to remain in the internet system. If this field contains the value zero, then the datagram must be destroyed. This field is modified in internet header processing. The time is measured in units of seconds, but since every module that processes a datagram must decrease the TTL by at least one even if it process the datagram in less than a second, the TTL must be thought of only as an upper bound on the time a datagram may exist. The intention is to cause undeliverable datagrams to be discarded, and to bound the maximum datagram lifetime." (S. 14)

Man sieht, diese Dokumente sind durchaus im Klartext lesbar.

Nun kommt das Feld, das die die Trennschicht zum darüberliegenden Protokoll markiert: Protocol. Dort steht, welches Protokoll sich um das Paket inklusive der transportierten Daten kümmern soll, wenn es am Bestimmungsort angekommen ist. Z. B. könnte das TCP sein, und das wäre der Fall, stünde da eine „6", also 0110. Es ist also grundsätzlich offen und regelbar, welches Protokoll die Arbeit an den Daten übernimmt. Hier wird die Verantwortung an die nächste Schicht abgegeben.

Header Checksum, Header-Prüfsumme, hilft dabei, Übertragungsfehler im Header zu entdecken. Auffällig ist zunächst, dass IP sich nur um den Header kümmert. Ob Fehler in der Datennutzlast aufgetreten sind, hilft dieses Feld nicht zu entscheiden, das ist Sache eines übergeordneten Protokolls, falls es überhaupt geschieht. Hier passiert etwas ziemlich Raffiniertes: Aus den Daten des Headers selbst wird die Prüfsumme berechnet, einer Quersumme ähnlich, die an jeder Station, die das Paket weiterreicht, mit den Daten des Headers abgeglichen werden kann. Die Prüfsumme ist gerade so gebaut, dass die Summe aller Header-Felder und des Prüfsummenfeldes genau eine Reihe von Einsen ergeben muss. Kommt das nicht so heraus, gibt es also einen Übertragungsfehler, wenigstens ein Bit ist umgesprungen.

Die Prüfsumme muss natürlich nach jedem *Hop*, jedem Weiterreichen von einer Station zur nächsten, neu berechnet werden, weil schließlich das TTL-Feld, Teil des Headers, sich ändert. Das Schöne ist, das das mit jeder Headerlänge funktioniert, weil ja alle Header-Bits miteinander verrechnet werden, und dass es ziemlich wahrscheinlich ist, einen Fehler zu entdecken.

2. IP-Adressen

Nun zur Hauptsache, zu den Adressen: im Feld Source Address steht der Absender, in Destination Address der Empfänger, beides in Form der IP-Nummern, vier Gruppen von Zahlen zwischen Null und 255. Etwa:

```
193.174.32.14
```

Dieses ist die IP-Adresse des Web-Servers der Leuphana Universität Lüneburg. Oder

```
128.30.52.54
```

als die IP-Adresse des World-Wide-Web-Consortiums www.w3.org.

Eine IP-Adresse besteht immer aus zwei Teilen, die der Adresse selbst nicht anzusehen sind: dem rechten Teil, der das Gerät in seinem Subnetz, das von der Organisation betrieben wird, in der das Gerät steht, bezeichnet, und einem linken, der das Subnetz adressiert. An der Grenze des Subnetzes steht immer ein Router, der also zwischen lokalem Netz und dem Internet vermittelt. Der allerorten gezogene Vergleich mit Anschluss (rechter Teil) und Telefonvorwahl (linker Teil) trifft die Sache ganz gut, denn innerhalb eines Ortkennzahlenbereichs, eines Subnetzes, kann man den globalen Teil ignorieren. Wenn

Sie Ihren Rechner schon einmal haben konfigurieren müssen, um eine Internet-Verbindung herzustellen, dann haben Sie mit dieser Unterscheidung schon einmal Bekanntschaft gemacht.

An dieser Stelle schlägt, wie so oft, die technoide Kultur des IT-Jargons durch: die Unterscheidung zwischen Subnetz und Internet wird durch eine Reihe von binären Einsen, gefolgt von binären Nullen, gezogen, der Subnetzmaske. Die Einser-Positionen markieren den Internet-Teil der Adresse, die mit den Nullen den Bereich der lokalen Subnetze, in denen jeweils genau ein Router steht:

```
11111111.11111111.11111111.00000000
```

oder dezimal

```
255.255.255.0
```

bedeutet dann: die linken drei Ziffernfolgen bezeichnen den globalen, die ganz rechten die lokalen Subnetz-Adressraum. Dort können also höchstens 256 Geräte adressiert werden, von 0, über 1, 2, ... bis 255.

Größere Subnetze haben dann Subnetzmasken, in denen mehr Nullen stehen: 255.255.0.0 etwa oder gar 255.0.0.0.

Wendet man die Subnetzmaske 255.255.255.0 auf 193.174.32.14 an, hieße das, dass jede Adresse eines Geräts im Subnetz mit 193.174.32 beginnen müsste, dass Adressen für diese Geräte dann alle die Form 193.174.32.0 bis 193.174.32.255 haben müssen. Und andererseits: Die Wegefindung im Internet muss sich nur um den globalen Teil kümmern, um die 193.174.32. Den Rest erledigt dann der letzte Router, der des Subnetzes. Für Netzwerkadministratoren bedeutet dies, dass sie sehr bedacht mit den Adressen umgehen müssen, dass die Anlage des Gesamtnetzes mit seinen Endgeräten und Routern eine delikate Angelegenheit darstellt.

Die Verteilung der globalen Internet-Anteile der IP-Adressen ist eine schwierige Angelegenheit. Für sie gibt es eine weltweit agierende Organisation die Internet Corporation for Assigned Names and Numbers (ICANN), eine private Stiftung, die aus dem US-amerikanischen Handelsministrium ausgegliedert wurde. Sie wurde anfänglich von Jon Postel und Vint Cerf geleitet, die wir schon als Internet-Pioniere kennen. Die ICANN ist mit internationalen Fachleuten besetzt. Sie regelt die Nummern- und die Namensvergabe für Internet-Adressen.

Exkurs: Domain Names

So nützlich die Zahldarstellung mit ihrer Aufteilung in Inter- und Subnetz auch sein mag, das ist nichts für Menschen, die sich eher Namen merken können. Allerdings ist diese menschliche Schwäche nicht von Anfang an problematisch gewesen, denn zunächst gab es ja nur wenige Hosts. Etwa vier. Als das Netz wuchs, kam man dann nicht mehr so recht hinterher damit, Buch über die Nummern zu führen, und die nahe liegende Lösung bestand darin, einen Dienst zu erfinden, der eine Nummern-Namens-Zuordnung automatisch übernähme. Heraus kam der Domain Name Service, **DNS**, der zwischen 1983 (RFC 882) und 1987 (RFC 1035) festgelegt wurde, und der geht so:

Ein User, also ein Programm, benutzt bei seinem Betrieb einen Namen, etwa www.w3.org. Damit die Anfrage an diese Stelle im Internet gelangt, muss daraus die IP-Adresse in Form der vierteiligen Ziffernfolge werden, denn die enthält viel mehr Strukturinformation und ist hierarchisch aufgebaut, von links nach rechts: 128.30.52.54. Man kann ihr etwa ansehen, wo der Host ungefähr steht, so, wie bei einer Telefonnummer mit Vorwahl.

Also muss in einer Datenbank nachgeschlagen werden, welche Nummer hinter dem Namen steckt. Am einfachsten wäre es, genau eine Datenbank zu haben – im Bilde: ein Telefonbuch – die immer in Anspruch genommen wird und alles weiß. Dieses ist aber in Hinblick auf das Wachstum und die Stabilität des Internet keine gute Lösung, und so hat sich die Idee durchgesetzt, ein hierarchisches System von Domain Name Servern aufzubauen, die befragt werden können. Diejenigen der höchsten Hierarchiestufe haben dann ein Verzeichnis von DNS auf niedrigerer Ebene, deren Adresse sie wenigstens mitteilen können, und dann wird das Programm hoffentlich dort fündig. Sie kennen den Witz über Dienstgrade sicher. Ein Stern auf der Schulterklappe bedeutet: Kann lesen. Zwei: Kann lesen und Schreiben, und drei: Kennt jemand, der lesen und schreiben kann.

Und in der Tat: das sich Durchfragen bis zu dem Server, der schließlich die Adresse hat, darf als Organisationsprinzip des DNS gelten. Es gibt dreizehn DNS Root Server, sie heißen A bis M und sind redundant und vervielfacht über die Welt verteilt, allerdings stehen die meisten der Ursprungs-Server in den USA:

(http://www.root-servers.org/)

Der erste, www.verisign.com, ist etwa nur für die Top-Level-Domains .com, .net, .edu zuständig, also für den Kommerz, das Netz selbst und die Bildung. Hier, bei den Internet Names, wird die Hierarchie anders herum praktiziert: ganz rechts stehen die Top-Level-Domains, TLD, die zunächst einmal Organisationstypen in den USA beschrieben hatten. Nach links wird es dann spezifischer, „www.verisign.com“ heißt dann: Verisign ist ein kommerzieller Betrieb, die Domain heit „verisign“ und das „www“ bezeichnet halt den WWW-Server dieser Domain.

Später wurde dann klar, das auch noch andere Länder auf der Erde Internt-Namen verwenden wollen. Dadurch kamen dann schließlich die Kürzel für die einzelnen Länder hinzu, von .ac (Ascension Island) bis .zw (Zimbabwe). Natürlich gibt es .at und .de, und eine ganz beliebte ist .tv, weil sie sich so nach Fernsehen anhört, aber dem Inselstaat Tuvalu gehört, einem der kleinsten Staaten der Erde, der zudem im Begriffe ist, mit seinen paar Riffs und Atollen der Klimakrise wegen im Pazifik zu versinken. Wenn früher Südseestaaten wunderschöne große Briefmarken produzierten und so mit Hilfe der Markensammler ihren Staatshaushalt aufbesserten, verkaufen heutige Paradiese ihre Top Level Domains an Fernsehleute.

Und wie sich das gehört für eine Kulturtechnik, die ganz in der Gesellschaft angekommen ist, gibt es Streit über die zulässigen länderunabhängigen TLD. Den trägt die ICANN aus, und das findet dann auch Interesse genug, um in den Feuilletons diskutiert zu werden.

Wenn man eine Domain einrichten will, die dann auch von den Domain Name Servern gekannt und in eine Nummer übersetzt werden kann und die auch eindeutig, also noch nicht vergeben, ist, wendet man sich an seine nationale Nummern- und Namens-Autoriät. Für Deutschland heißt diese DENIC (Deutschland-Network Information Center), und nach eigenem Bekunden (www.denic.de) betreibt sie ein Netz von Name-Servern, die für die Top Level Domain .de dann Auskunft über alle engemeldeten Domains geben können. Die Root-Server müssen dann nur die Adresse der DENIC-Server kennen, die haben dann die Details. Österreichs NIC heißt NIC.at. Für eine Jahresgebühr von etwa 120 Euro kann man dann diese Domain benutzen.

Das NIC.at hat ein kleines Filmchen darüber gemacht, wie ein ganz normaler Internet-User, der allerdings einem Insassen des Computerspiels The Sims erschreckend ähnlich sieht, die IP-Adrsse hinter einem Namen aufdeckt:

Womit der Exkurs beendet sein soll.

3. IP-Adressen sind knapp

Mit vier Ziffenfolgen zwischen 0 und 255 kann man insgesamt 2^{32} verschiedene Adressen schreiben, das sind gut vier Milliarden, eine Milliarde je Anfangs-Standort des ARPANET, aber nicht genug für alle sechseinhalb Milliarden Menschen auf der Erde. Ein sparsamer Umgang mit dem Adressraum ist also angezeigt, und so gibt es zwei kleine Lösungen, die für das IP Version 4, und die große, das IP Version 6.

Kleine Lösung Nr. 1: DHCP

Da nicht immer alle Nutzerinnen und Nutzer des Internet zugleich online sind, werden viel weniger Nummern tatsächlich gleichzeitig verwendet, als es User gibt. Deshalb vergibt der Internet-Dienst Dynamic Host Configuration Protocol (**DHCP**) Nummern aus Adressräumen nach Bedarf. Man geht online, und DHCP gibt einem eine aktuell gültige IP-Adresse und stellt auch ein, welche Nummern der nächste Router und der DNS-Server haben, die für einen zuständig sind. Die Netzwerk-Administratorinnen und -Administratoren pflegen die IP-Adressen ein, die von DHCP vergeben werden dürfen. Dadurch geht der Zusammenhang zwischen dem einzelnen Computer und der IP-Adresse verloren, aber es werden Adressen gespart.

Kleine Lösung Nr. 2: NAT

In einem Privathaushalt gibt es eine kleine, aber nicht genau bekannte Zahl von Nutzerinnen und Nutzern, für diesen ist ein Internet Service Provider, der typischer Weise ein Telekommunikationsunternehmen ist, zuständig. Diese Internet-Konsumentinnen und -Konsumenten haben typischer Weise eng umrissene Bedürfnisse, wollen das WWW benutzen, E-Mails schreiben und senden, aber keine eigenen Server betreiben. Für diesen Zweck kann man sogar das Prinzip aufgeben, dass jeder ans Internet angeschlossene Computer eine unverwechselbare individuelle IP-Adresse hat. Der Internet Service Provider teilt dann jedem Haushalt zu einer festen und unverwechselbaren IP-Adresse einen IP-Nummernbereich zu, der sich in einem anderen Privathaushalt sogar wiederholen darf. Diese Technik heißt Network Address Translation (**NAT**), und sie erfordert natürlich eine saubere Vergabestrategie. Im RFC 1918 – Address Allocation for Private Internets – sind dafür Nummernbereiche reserviert, und der für

typische Haushalte eingesetzte beginnt mit einer 10, die weiteren Nummern werden dann dynamisch vergeben.

Während dieser Text geschrieben wird, hat der Rechner des Autors die IP-Adresse 10.0.0.19. Die Teilnetzmaske ist 255.255.255.0, und der Router, der die Internet-Pakete vermittelt, hat die Adresse 10.0.0.138. Alles diese sind mehrdeutige Adressen, die andere in anderen Haushalten auch haben. Aber das spielt bei einer so schlichten konsumptiven Nutzung wie der eines Standard-Internet-Nutzers keine Rolle, lässt sich auf die eine feste IP-Adresse abbilden.

Die große Lösung: IPv6

Die große Lösung besteht darin, im Zuge des **Internet-Protokoll**s mit der **Versionsnummer 6** sehr viel längere IP-Adressen zuzulassen, die nach menschlichem Ermessen tatsächlich auf absehbare Zeit ausreichen werden. Die Adressen sind bei IPv6 128 Bit lang, was 2^{128} Adressen erlaubt, das sind $3*10^{38}$ Stück oder fast 10^{29} pro lebendem Menschen auf der Erde, eine Eins mit 29 Nullen dahinter.

In RFC 2460 ist IPv6 bereits in 1998 beschrieben worden, seit Beginn der 1990er Jahre ist es geplant worden, aber konnte sich bislang nicht flächendeckend durchsetzen. Dies liegt vor allem an alter Infrastruktur die noch immer in großer Zahl in Betrieb ist. Ein Machtwort wie das der ARPA in 1983, das den Übergang zu TCP/IP erzwang, ist von niemandem zu erwarten. Wahrscheinlicher ist es wohl, dass neue Gerätetypen wie Internet-taugliche Mobilfunkgeräte, die IPv6 implementiert haben, der Entwicklung Schwung verleihen werden. In der Zwischenzeit kann man sich behelfen, etwa dadurch, dass veraltete Hard- und Software, die nur IPv4 beherrscht, „durchtunnelt“ (Kurose 399 ff), also als minderwertiges Transportmedium benutzt wird.

4. Routing

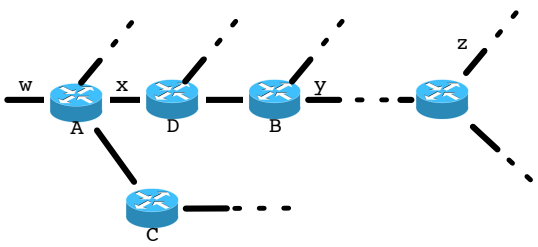
Wie nun vollbringt das technische System des Internet das Kunststück, alle die astronomisch vielen Datenpakete durch das Gewirr des Netzes zu leiten, wie funktioniert das Routing? Vor allem: wie geht das, wo doch ständig neue Server hinzukommen, Leitungen zusammenbrechen, neue Verbindungen gezogen werden?

Das Prinzip der Wegefindung beruht auf einer lokalen Auskunft, in welche Richtung, d. h. zu welchem nächsten Router ein Paket mit einer bestimmten Empfängeradresse reisen soll. Liest man nämlich die Empfängeradresse im IP-Header von links nach rechts, sieht man ja das Teilnetz, zu dem der adressierte Rechner gehört, unter den Einsen der Subnetzmaske. Die ganz grobe Richtung, das größte dem Router bekannte Teilnetz, in das das Paket soll, steckt in der hierarchisch gebauten IP-Adresse, es ist die kürzeste dem Router bekannte Nummerkombination, gelesen von links. Ein Router reicht es zum Router des Teilnetzes weiter, dieser muss dann selbst wissen, was er damit anstellt.

Dieses wird wiederholt, bis der Router des letzten Teilnetzes dann ganz konkret die detaillierte volle IP-Adresse kennt und auch das Paket direkt zum Empfänger adressieren kann.

Der Vergleich mit dem Telefonnetz mag hier nützlich sein, denn eine Telefonnummer besteht auch aus mehreren Teilen, auch von links nach rechts vom Allgemeinen zum Konkreten gehend. Ganz links die nationale Vorwahl: 0049 etwa für Deutschland. Dann die Vorwahl des Ortsnetzes: 04131 für Lüneburg. Schließlich der Anschluss: 6771202. Das Ganze gibt dann: 0049.4131.6771202.

Ein Router erledigt die Wegeleitung auf Grundlage einer lokal gespeicherten Tabelle, die für ein angrenzendes Teilnetz die Richtung und die Entfernung angibt, die der Router bis dahin weiß.



Router D habe folgende (abstrakt gefasste) Routingtabelle (nach Kurose S. 426):

Zielsubnetz	nächster Router	Hops bis zum Zielnetz
w	A	2
y	B	2
z	B	7
x	–	1
…	…	…

Um ein Paket aus einem Subnetz, das zu D gehört, nach w zu schicken, ist der nächste zuständige Router links von ihm A, und das ist der zweite, mithin macht das Paket zwei Hops. Nach y sendet D ein Paket über B, und das macht auch zwei Hops. Will aus einem Teilnetz, das direkt an D angeschlossen ist, jemand ein Paket nach z schicken, tut er das auch über den Router B und braucht, so der Kenntnisstand von D, sieben Hops, die im Diagramm durch die gepunkteten Linien angedeutet sind. B hat dann eine eigenen Routingtabelle, und in der dürfte es einen Router geben, der rechts neben ihm ist, und über den B für das Erreichen von z nur noch sechs Hops angibt.

Diese Routingtabelle ist anfänglich von einer Netzwerkadministatorin oder einem Netzwerkadministrator aufgebaut worden, die oder der das autonome System, das Teilnetz, genau kennt, also weiß, zu welchem nächsten Router man ein Paket schicken muss, damit es in die richtige Richtung wandert.

Die IP-Adresse des Paketempfängers wird nun mit der Liste der Zielsubnetz-Adressen abgeglichen. Wird das Teilnetz gefunden, mit dem der Router etwas anfangen kann, wird nachgeschaut, welcher der nächste Router ist, der mit den wenigsten Hops zu erreichen ist. Dort hin sendet der Router das Paket.

So weit, so gut, solange sich nichts ändert, beschreibt eine solche statische Routingtabelle alles völlig hinlänglich. Nun gibt es aber Dynamik:

Die Router sind untereinander in regelmäßigem Kontakt. Spätestens alle 30 Sekunden schickt ein Router allen seinen Nachbarn die Nachricht, dass es ihn noch gibt, dass er intakt ist. Er schickt aber auch dann, wenn er Nachricht davon bekommt, dass an den Weiterleitungspfaden sich etwas geändert hat, dann auch diese Information an seinen Nachbarn weiter. Das kann etwa passieren, wenn eine neue Verbindung gezogen wurde, die bislang dem Router noch nicht bekannt war.

So könnte es passieren, dass Router A eine solche bessere Route nach Teilnetz z über den Router C entdeckt hat, in vier Hops. Dann schickt er an D seine neue Routingtabelle, und die sieht so aus:

Routingtabelle A

Zielsubnetz	nächster Router	Hops bis zum Zielnetz
z	C	4
w	–	1
x	–	1
…	…	…

Aus der ersten Zeile schließt nun Router D messerscharf, dass es eine kürzere Route zum Teilnetz z geben muss als die, die er kennt – über B mit 7 Hops –, wenn es von A aus über C nur 4 Hops entfernt

ist, und A in einem Hop erreichbar, ist die Route über A nun also fünf Hops lang und damit kürzer als die alte.

Also korrigiert er seine Routingtabelle folgendermaßen, indem er die kürzere Route hinzufügt:

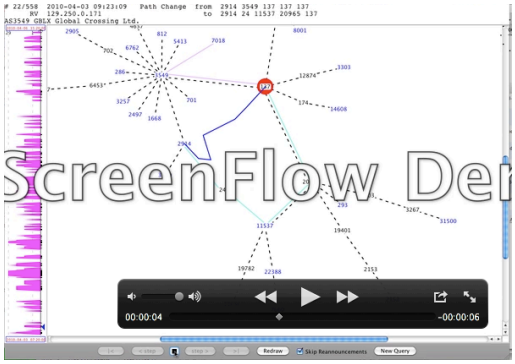
neue Routingtabelle D

Zielsubnetz	nächster Router	Hops bis zum Zielnetz
w	A	2
y	B	2
z	A	5
z	B	7
x	–	1
…	…	…

Dadurch hat sich die Wegeleitung verändert, sie kann nun auf einen neu entdeckten Pfad reagieren, und solange es die neue Route gibt, wird D sie vorschlagen, weil sie kürzer ist als die über B.

Wenn sich C bei A nicht mehr meldet, weil er z. B. defekt ist oder die Leitung gekappt, streicht A die Route nach x über C wieder, schickt dies seinem Nachbarn D, und der entfernt ebenfalls diese Möglichkeit aus der Tabelle. So wäre auf diese Netzwerkstörung reagiert worden. Und das war die ursprüngliche Idee der Militärs, dass Störungen automatisch erkannt und umgangen werden können müssen.

Schauen wir uns einmal die zeitliche Entwicklung der Routingtabellen eines Routers an! Unter http://bgplay.routeviews.org/bgplay/ kann man sich für eine bestimme IP-Adresse die Tabelleneinträge als gezeichnete Pfade anschauen. Es ist dies ein Mitschnitt des Routers 193.204.0.0/15. Er bedient die fünfzehn linken Ziffern seiner IP-Adresse, binär aufgeschrieben, gezeigt wird, wie sich eine Route ständig ändert zwischen dem 4. und dem 6. April 2010:



Ein Router weiß nicht alles über das gesamte Internet. Er hat keinen Plan aller IP-Adressen der ganzen Welt. Er weiß nur alles über das Subnetz, dessen Teil er ist, über das so genannte „autonome System“. Die Router unterscheiden sich darin, wie detailliert ihre Kenntnis der Teilnetze ist. Die ganz großen kennen auch nur die ganz großen Verbünde, die Zielsubnetze werden dann nur mit den ersten Ziffern der IP-Adresse bezeichnet sein, die Teilnetzmaske hat nur wenige Einsen und viele Nullen, es stecken dann Teilnetze mit selbst sehr vielen einzelnen IP-Adressen dahinter. Die Router der kleinen Subnetze kennen dann alle die IP-Adressen ihres Subnetzes und können direkt zustellen. Es gibt also ein System von Subnetzen, manche kennen die großen anderen Subnetze, mache kennen die vielen IP-Adressen des eigenen kleinen Zuständigkeitsbereichs. Diese Aufteilung eines Netzes in wenige sehr wichtige Oberzenzentren und viele weniger wichtige Enknoten ist ein Charakteristikum solcher Netze wier dem Internet, man nennt sie skalenfreie Netze.

Nun besteht das gesamte Internet aber aus lauter autonomen Systemen. Eines ist das Universitäts- oder Firmennetzwerk, das grenzt an den nationalen Internet-Provider, der selbst ein autonomes System be-

treibt, das man oft als Backbone bezeichnet. Auch dieses System, das Backbone, hat vollständige Information über seinen eigenen Aufbau und den rechten Weg eines Pakets quer durch das Netz seiner Router.

Und jetzt werden die Nachbarschafts-Verhältnisse wichtig, denn ein autonomes System kann seinem Nachbarn mitteilen, wie man Pakete richtig durch die Gegend schickt, so weit es selbst davon weiß. Wenn das gesamte Internet auf diese Weise aus einem Netz benachbarter autonomer Systeme besteht, und genau das tut es, dann kann über Austausch an den Grenzen, so zu sagen als Plausch über den Gartenzaun, sich die Kenntnis darüber verbreiten, wie man von einer Stelle zur nächsten kommt.

Schauen wir uns an, wie ein Datenpaket von der Leuphana Universität Lüneburg zum WWW-Server der Uni Wien reist. Man kann nachvollziehen, welche Stationen es nimmt, wenn man ein Internet-Programm namens „Traceroute“ verwendet:

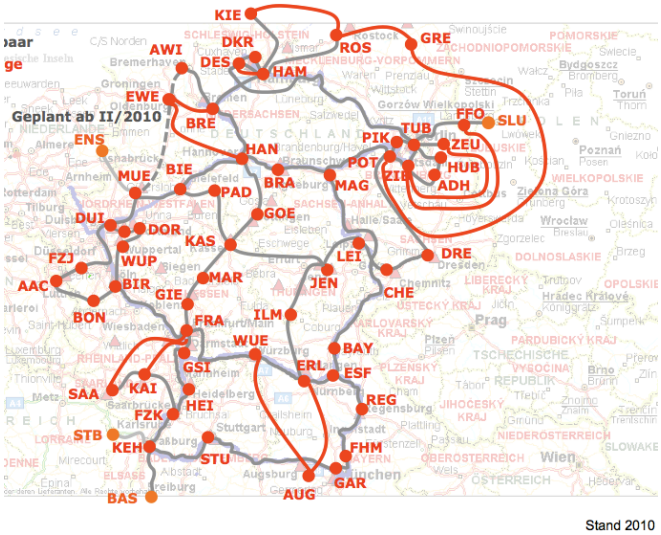
```
traceroute to www.univie.ac.at (131.130.70.8)

1   141.39.208.196 (141.39.208.196)
2   193.174.34.103 (193.174.34.103)
```

Das ist der Router an der Universität Lüneburg, der direkte Verbindung zum Internet Service Provider der deutschen Universitäten hat, dem DFN-Verein. Er wurde vom messenden Rechner aus über 141.39.208.196 im autonomen System der Leuphana erreicht. Es wird also eine letzte Zeile, eine Art Lumpensammler für alles, was davor innerhalb des Autonomen Systems nicht adressierbar war, in der Routing-Tabelle des zuständigen Routers an der Leuphana gegeben haben, die heißt:

0.0.0.0 (Teilnetzmaske 255.255.255.255) 193.174.34.103 2. Schicke alles andere also nach 193.174.34.103, und so braucht es zwei Hops in das Internet.

3 xr-ham1-ge9-10.x-win.dfn.de (188.1.230.17)

Hamburg ist die nächste Station im Subnetz des DFN-Vereins.



Stand 2010

4 xr-bre1-te2-1.x-win.dfn.de (188.1.144.90)

Dann Bremen im DFN.

5 zr-han1-te0-0-0-3.x-win.dfn.de (188.1.144.85)

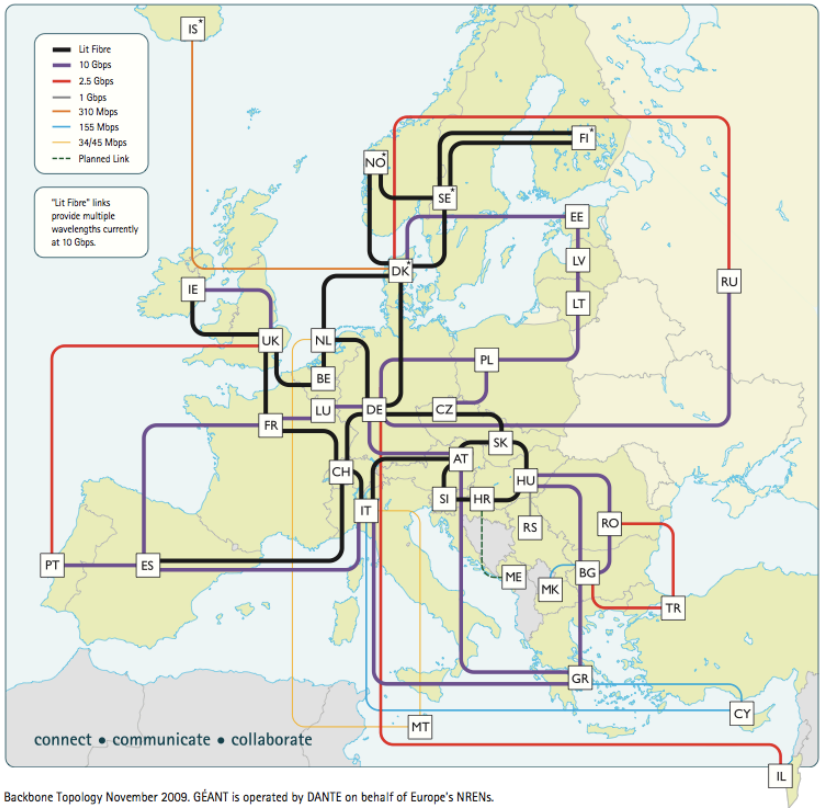
Hannover im DFN.

6 zr-fra1-te0-0-0-4.x-win.dfn.de (188.1.145.213)

Frankfurt im DFN.

7 dfn.rt1.fra.de.geant2.net (62.40.124.33)

Hier ist das Paket schon im Geant-Backbone:



Backbone Topology November 2009. GÉANT is operated by DANTE on behalf of Europe's NRENs.

(http://www.geant.net)

8 so-6-0-0.rt1.pra.cz.geant2.net (62.40.112.37)

Hier in Prag, Geant.

9 so-7-2-0.rt1.vie.at.geant2.net (62.40.112.5)

Hier im Geant-Netz in Wien.

10 aconet-gw.rt1.vie.at.geant2.net (62.40.124.2)

11 vlan72.wien2.aco.net (193.171.23.21)

Hier hat es Geant verlassen.

12 ares.cc.univie.ac.at (193.171.13.2)

Und ist hier an der Universität Wien angelangt.

Die Routing-Tabellen der beteiligten Server haben der IP-Adresse 131.130.70.8 der Uni Wien angesehen, zu welchem Subnetz sie jeweils gehört:

Der Router in Lüneburg hat gesehen, dass die Adresse nicht zu Lüneburg gehört und das Paket deshalb an den DFN-Router nach Hamburg gesendet. Der hat es über Bremen und Hannover nach Frankfurt geschickt, weil es dort einen Übergabepunkt zum Geant-Netz gibt, das ganz Europa bedient. Das weiß, das es ein solches Paket am besten über Prag nach Wien schickt und beim Österreichischen Provider abgibt, der wieder die Uni Wien kennt und es dort lässt.

Für jedes dieser Subnetze waren es nur eine Handvoll Hops, die es selbst kontrollieren kann, bis es sie ans nächste Subnetz abgibt. Und ändert sich etwas an der Netz-Topologie, sorgen die Nachrichten an den jeweiligen nächsten Nachbarn für eine angemessene Reaktion auf die Veränderungen, seien es neue Routen oder gekappte Leitungen.

5. TCP: die Verbindung im Internet

Die Geschichte des Internet weiß von Kämpfen zwischen den Telekommunikationsunternehmen und den akademisch orientierten Internet-Pionieren. Es ging um die Frage, wo die Raffinesse stecken solle, im Netz oder in den Endgeräten. Die Telekommunikationsunternehmen waren es gewohnt, und es hätte ihren Interessen entsprochen, das Netz unter Kontrolle zu behalten und die Endgeräte als bloße Konsumenten zu behandeln. Dies war aber weder im Interesse der Akademiker noch des Militärs, und auch die aufkommende populäre PC-Kultur trug dazu bei, dass Diversität und Heterogenität die Oberhand behielten und es nicht zu einem monolithischen Netz kam.

Im Protokollpaar TCP/IP, das für diese Heterogenität stand, waren die Gewichte also so verteilt, dass das Netz – bei uns: das Internet-Protokoll IP – schlicht angelegt waren und die Verantwortung für das Zustandekommen einer verlässlichen Verbindung auf Seiten der Endgeräte lag. Für die heutige Zeit heißt das, dass selbst Endgeräte wie Internet-fähige Mobiltelefone anspruchsvolle Dienste abwickeln können, etwa einen Web-Server. In der Vorstellung der Telekommunikationsanbieter wären sie dumme Endgeräte geblieben. Aber zum Glück kam es anders, also muss ein weiteres Protokoll für die verlässliche Ende-zu-Ende-Verbindung sorgen: TCP, das Transmission Control Protocol, und das wirkt in den Endgeräten, und nur in ihnen, die Router wissen nichts davon.

Auf Grundlage von IP, das weder für die Integrität der Datennutzlast noch überhaupt für das Zustandekommen einer Verbindung garantiert, das einen unzuverlässigen so genannten *best-effort*-Dienst realisiert, also sein Bestes tut, regelt TCP einen zuverlässigen Datentransfer. In Fällen, wo es eher auf Durchsatz als auf Sicherheit ankommt, etwa bei der Internet-Telefonie oder der Übertragung von Videos, verzichtet man auf diese Sicherheit und setzt das unzuverlässige UDP, User Datagram Protocol, ein, wenn es nicht so schlimm ist, wenn ein paar Einzeldaten fehlen, im Vergleich dazu, dass es lange Aussetzer gäbe.

TCP hingegen hat zwei Aufgaben, nämlich erstens dafür zu sorgen, dass der Datenverkehr zuverlässig abläuft, also die Datennutzlast tatsächlich, unverfälscht und in der Reihung, wie sie angeschickt wurde, auch ankommt, und zweitens, dass die Endgeräte immer angemessen mit Daten versorgt werden, es regelt also durch raffinierte Maßnahmen eine Flusskontrolle.

Schauen wir uns an, wie TCP einen zuverlässigen Datentransfer ins Werk setzt! Das Transport Control Protocol ist in den RFC 793 (1981) und RFC 1323 (1992) beschrieben worden. Zuverlässig wird man die Datenübertragung dann nennen können, wenn die abgesandten Daten unverändert, ohne Lücken und ohne Duplikate beim Empfänger ankommen (Kurose S. 284 ff). Da IP die Daten in Form einzelner Pakete durch die Netze navigiert, geht es also so gut wie immer um eine ganze Reihe von Segmenten, aus der der Datenstrom besteht, der hier zuverlässig zu übertragen ist.

Der Datentransfer findet in drei Schritten statt:

Schritt 1: Begrüßung

Der Sender schickt eine Verbindungsanfrage an den Empfänger und eine zufällig gewählte initiale Sequenz-Nummer, die den Start der laufenden Nummer für die vom Sender abgeschickten Daten bildet.

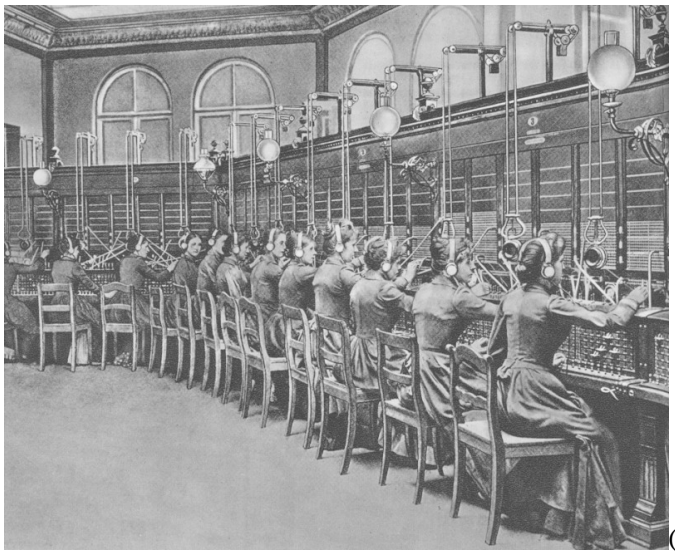
Der Empfänger akzeptiert die Anfrage, richtet sein Paket-Eingangslager, den Puffer, ein, und sendet seinerseits eine zufällig gewählte Empfänger-Sequenznummer zurück, mit der beginnend seine Rückmeldungen dann durchnumeriert werden. Er schickt die um Eins heraufgezählte Sender-Nummer mit, die klar macht, dass er auf die gerade empfangene Botschaft des Senders reagiert hat, dort also anschließt.

Es ist besser, dass beide nicht bei Null mit der Zählung beginnen, denn das täten ja alle, und Verwechselungen wären unvermeidlich.

Der Sender schickt eine Nachricht, dass er die Bestätigung des Empfängers erhalten hat, zählt dessen laufende Nummer hoch, schließt also bei seiner Nachricht an, und benutzt die gerade vom Empfänger hochgezählte Sendernummer, denn es ist ja tatsächlich das zweite Paket, das er schickt. Der Sender richtet seinerseits ein Paket-Ausgangslager, einen Puffer, an.

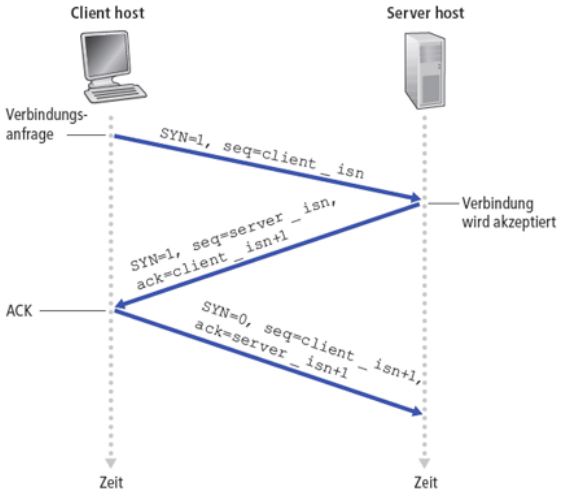
Für den Sender und den Empfänger wird der Datenstrom mit einer eindeutigen Nummer gekennzeichnet, die sich aus der IP-Adresse des Kommunikationspartners und dem Dienst zusammensetzt, der benutzt wird, etwa E-Mail, Web, FTP. Es hat sich hierfür die Metapher der Steckdose etabliert, dieses Kennzeichen heißt Socket.

Mir kommt die Assoziation des Fräulein vom Amt, das den Kontakt der Gesprächspartner durch Einstöpselung in Kontakte herstellte.



(http://www.s-storbeck.de/cms/images/stories/altetelefone/vermittlungstechnik/pics_gross/manuell_stettin.jpg)

In einem Zeitablauf-Diagramm sieht das folgendermaßen aus:



(Kurose S. 296)

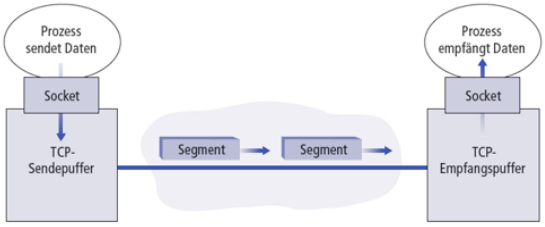
Die anfänglich zufällig gewählten und dann jeweils von der Gegenseite hochgezählten Nummern bieten die Grundlage dafür, dass die zwischen Sender und Empfänger ausgetauschten Nachrichten eine ununterbrochene Kette bilden, wie beim Gänsemarsch.

(http://de.wikipedia.org/w/index.php?title=Datei:Bundesarchiv_Bild_183-87238-0002,_Henningsleben,_Kindergartenkinder_im_Spiel.jpg&filetimestamp=20081204160721)

Der Gänsemarsch von jeweils vom Anderen weiter gezählten Positionen im Datenstrom zwischen Sender und Empfänger bildet die Struktur für die Nachrichtenkette während der Begrüßungssequenz und auch danach. Stimmt die Numerierung nicht, ist die Kette unterbrochen, müssen Maßnahmen getroffen werden, die den Übertragungsfehler beheben.

2. Schritt: Übertragung des eigentlichen Datenstroms

Im besten Falle sendet der Sender, der Empfänger empfängt, der Empfänger bestätigt den Erhalt der Daten, der Sender reagiert mit dem Versand der nächsten Segmente und immer so weiter, bis alles angekommen ist. Der Datenstrom und auch die Serie der Bestätigungen bilden jeweils eine ununterbrochene Kette. Eine zuverlässige Datenübertragung wäre realisiert.



(Kurose S. 275)

Die zu übertragenden Daten werden in Segmente aufgeteilt, die die Datennutzlast ausmachen, wir haben es ja schließlich mit Paketvermittlung zu tun. Da diese Segmente unterschiedlich lang sein können, zählt TCP die Bytes ab, aus denen der Datenstrom besteht, der auf der anderen Seite rekonstruiert werden soll. Jedes TCP-Segment enthält eine Sequenznummer, und das ist nicht einfach nur die Nummer des gesendeten Segments, sondern die Nummer des ersten im TCP-Segment enthaltenen Bytes in der Abfolge aller Bytes, die den Datenstrom ausmachen, seine Position. Kommt das Segment gut an, kann es also sofort in den Puffer beim Empfänger an der richtigen Stelle einsortiert werden.

Der Empfänger sendet als Bestätigungs-Nummer die Position des nächsten erwarteten Bytes im Gesamt-Byte-Strom, teilt also mit, was er nun empfangen will und wohin er die Daten in seinen Puffer einzusortieren beabsichtigt. Der Sender schickt dann Daten ab dem erwarteten Byte, und so geht es immer weiter.

RFC 793 beschreibt es so: „Die Übertragung wird durch Verwendung von Sequenznummern und Bestätigungen verlässlich. Im Prinzip wird jedem Byte der Daten eine Sequenznummer zugewiesen. Die Sequenznummer des ersten Daten-Bytes eines Segments wird mit diesem Segment geschickt und heißt die Segment-

Sequenz-Nummer. Die Segemente tragen auch eine Bestätigungs-Nummer, die die Sequenznummer des nächsten erwarteten Bytes der Übertragung in umgekehrter Richtung ist.“ [1]

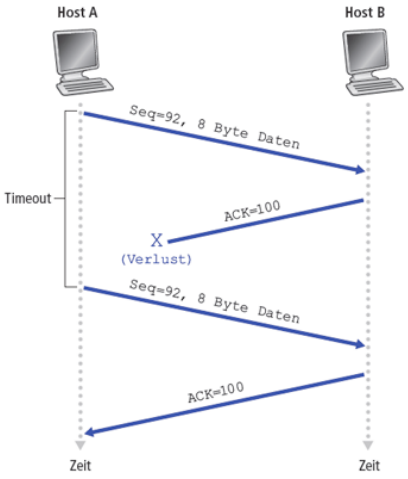
In einer besten aller Netz-Welten wäre TCP mit seiner Kontrolle überflüssig. Also müssen wir die Schikanen bedenken, die beim Versenden von Daten über den unsicheren Kanal IP den Datenstrom betreffen können. Und die wären:

Ein Paket kommt nicht oder verspätet oder beschädigt beim Empfänger an. Der Empfänger verschickt keine Empfangsbestätigung, schickt sie zu spät oder die Meldung ist beschädigt. Oder die Puffer drohen überzulaufen.

Da IP ja nicht dafür sorgt, dass ein Paket ankommt, muss es eben noch einmal gesendet werden, wenn es nicht korrekt angelangt ist. Das betrifft sowohl den eigentlichen Datenstrom wie auch die Empfangsbestätigungen, und das unter Bedingungen knapper Ressourcen, denn sowohl die Puffer sind nur endlich groß, und die Übertragungs-Bandbreiten auch.

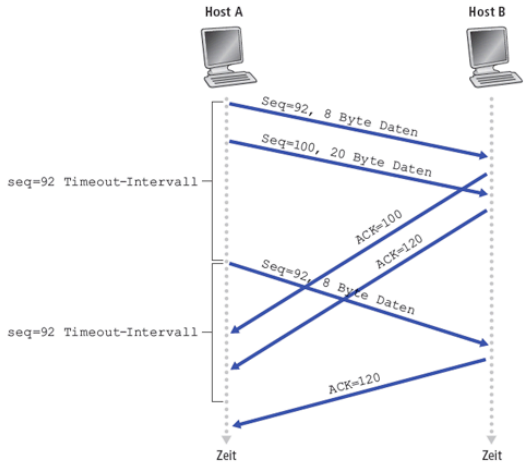
Klug, wie die Pioniere des Internet waren und sind, haben sie die Mechanismen, die im Fehlerfalle greifen sollen, den Implementationen des Dienstes überlassen, so dass es aktuelle, den jeweiligen technischen Gegebenheiten angepasste Lösungen geben kann. Es ist dies ein Motiv der Offenheit und des Kontrollverzichts, der meiner Ansicht nach überhaupt den Wesenszug des Internet ausmacht, sogar in einem Protokoll, das für Kontrolle verantwortlich ist.

Schauen wir uns drei Szenarien an, die Übertragungsprobleme und das Verhalten von Sender und Empfänger darstellen!

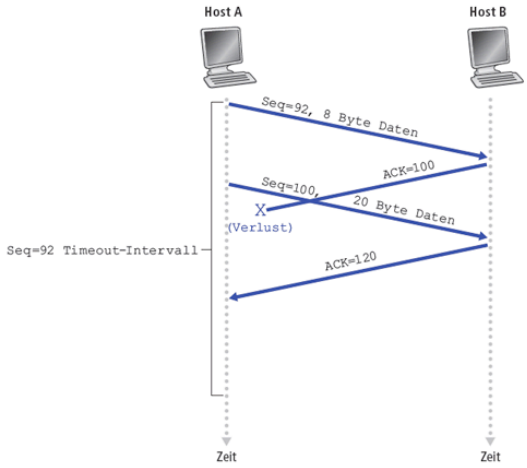


Kurose S. 286: ein Segment geht verloren

[1] „Transmission is made reliable via the use of sequence numbers and acknowledgments. Conceptually, each octet of data is assigned a sequence number. The sequence number of the first octet of data in a segment is transmitted with that segment and is called the segment sequence number. Segments also carry an acknowledgment number which is the sequence number of the next expected data octet of transmissions in the reverse direction.“ RFC 793 S. 10.



Kurose S. 287 Abb. 3.35: ein ungeduldiger Sender



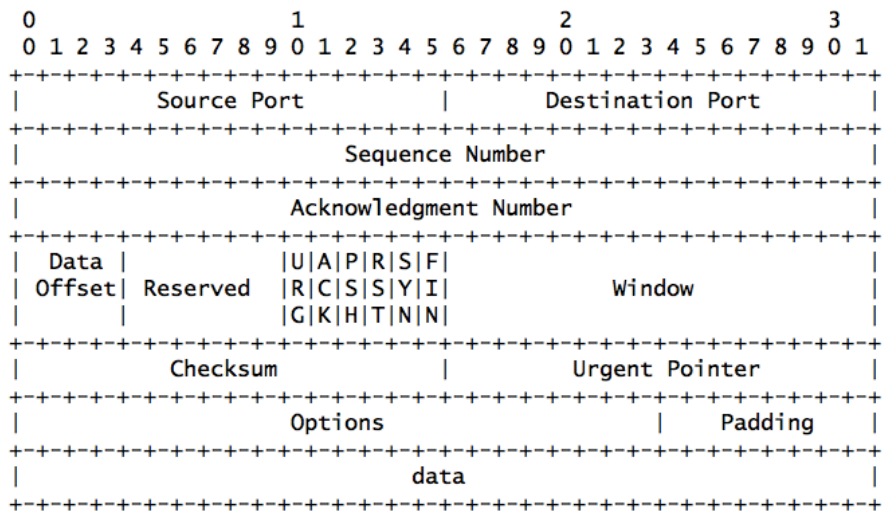
Kurose S. 288 Abb. 3.36: kumulative Bestätigungen – ein Empfänger denkt mit

Über die Flusskontrolle während der Datenübertragung soll hier nur gesagt werden, das es sich dabei um die Beobachtung der Sender- und Empfänger-Puffer handelt und eine angemessene Reaktion auf Überläufe und Entleerung. Die technische Lösung ist die eines Fensters zulässiger Segmente, so dass weder weit nach vorn noch weit zurück in die Vergangenheit gegriffen wird, damit die Puffer nicht überlaufen. Wer je eine Flüssigkeit durch einen Trichter in eine Flasche abgefüllt hat, weiß, dass das nicht einfach ist. Wenn der Trichter überzulaufen droht, muss man den Zufluss drosseln, man gießt heftiger, wenn der Trichter trocken fällt. Und man muss mächtig Obacht geben, dass die Flasche nicht überläuft. Der erlaubte Trichter-Füllstand entspricht dem Datenfenster, das für die Flusskontrolle vereinbart wird. Wer immer dachte, Netzwerktechnik sei eine trockene Angelegenheit, sollte diese Haltung vielleicht überdenken.

Dritter und letzter Schritt: Abbau der Verbindung

Der Sender schickt eine Ende-Meldung, der Empfänger bestätigt sie. Der Empfänger sendet eine Ende-Nachricht, der Sender bestätigt sie. Darauf hin können Sender und Empfänger ihre Puffer auflösen, und die TCP-Sitzung ist beendet.

Aufbau eines TCP-Segments



TCP Header Format

Ein TCP-Segment besteht aus folgenden Feldern:

Den Port-Nummern des Sende- und des Empfänger-Prozesses. Auf jedem der Rechner, die Daten über das Internet schicken, können verschiedene Prozesse laufen: E-Mail, Web, FTP oder andere. Damit die Beteiligten wissen, zu welcher logischen Verbindung das Segment gehört, gibt es Nummern für Prozesse. So hat das Web etwa die Portnummer 80. Zusammen mit der jeweiligen IP-Adresse, an die die Portnummer gehängt wird, wird eine eindeutige Identifikation des Datenstroms auf beiden Seiten hergestellt, der Socket, die Steckdose. Dieses markiert auch den Anschluss an die darüber liegende Protokollschicht.

Danach kommen die Sequennummern für den Gänsemarsch. Es folgen die Informationen dazu, wo die Daten beginnen, die Felder für die jeweiligen Phasen der Verbindung, für die Kennzeichnung, ob Verbindungsanfragen gestellt werden, Daten oder Bestätigungen übertragen oder das Übertragungsende vereinbart wird.

Window beschreibt den Ausschnitt aus dem Datenstrom, den die Beteiligten gerade vom anderen akzeptieren, damit ihre Puffer nicht überlaufen.

Die Prüfsumme schützt den Header und die Datennutzlast. Die restlichen Felder dienen dem Aushandeln des Verbindungsaufbaus zwischen den Partnern, manche sind auch bereits aus der Mode gekommen (Genaues bei Kurose S. 276 f), ich will auf die Details hier nicht eingehen.

Wie schon beim IP fehlen Vorkehrungen für Absicherungen gegen Fälschungen, es sind keine Felder für kryptographische Schlüssel vorgesehen.

Für die Diskussion im Rahmen dieser Darstellung ist wichtig festzuhalten: TCP errichtet verlässliche Verbindungen auf unsicheren Kanälen. Es tut dies in Eigenregie der Endgeräte, und das sind immer zwei. Das Prinzip besteht aus dem wechselseitigen Abzählen der zugesandten Nachrichten so, dass Lücken oder Doppelungen erkannt und durch erneutes Senden oder Löschen behoben werden können. Für die oberhalb von TCP liegenden Schichten heißt das, dass sie sich nun auf einen Übertragungskanal verlassen können, der für die Integrität des Datenstroms sorgt, allerdings nicht für einen Datendurchsatz. TCP realisiert damit eine logische Verbindung zwischen Kommunikationspartnern, ähnlich einer direkt geschalteten Leitung, allerdings ohne eine Bandbreitengarantie, wie eine solche sie böte.

Wichtigste Themen dieses Abschnitts:

- sauberer Aufbau der Internet-Protokolle in Schichten
- nur sehr wenige Annahmen über die jeweils darüberliegenden Schichten
- keine Verschlüsselung, keine Abrechnung eingebaut

- IP: vernetzt Netze
- TCP: simulierte Dauerverbindung