Evolutionen und Revolutionen im bisherigen Internet

Rolf Siebachmeyer
Betreuer: Nils Kammenhuber
Seminar Future Internet WS09/10
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: rolf@siebachmeyer.de

***Kurzfassung*— Die Geschichte des Internets ist durchsäht von kleineren und größeren Veränderungen. Nicht nur technische Grundlagen, auch Anwendungen, das soziale Verhalten im Netz und die Kommerzialisierung (um nur Einige zu nennen), haben sich seit Beginn es Internetzeitalters stark gewandelt. Diese Arbeit befasst sich in erster Linie mit einigen ausgewählten technischen Änderungen und der Frage, warum diese durchgeführt wurden.**

***Schlüsselworte*— ARPAnet, IP, IPv4, IPv6, Hosts-Datei, DNS, Class-A/B/C-Netze, BGP, CIDR, NCP, TCP, TCP-Congestion-Control.**

I. Einleitung

Rückblickend hat das Internet seit seiner Entstehung im Jahr 1969 einen stetigen Wandel durchlebt. Selten gab es tiefe Einschnitte und Veränderungen in die zugrundeliegenden Konzepte und Technologien. Diese Ausarbeitung befasst sich sowohl mit den Evolutionen des Internets, also dem stetigen Wandel, als auch mit den Revolutionen in der Internettechnologie, also den schlagartigen Veränderungen.

Durch das rasante Wachstum wurde bereits sehr früh klar, dass mit dem Internet ein Schneeball ins rollen kam, der eine ganze Lawine auslösen wird. Exponentielles Wachstum (siehe Abbildung 1) steigerte die Anzahl der Host von gut 200 im Jahr 1981 auf mehr als 1.000 bis Ende 1984. Weitere drei Jahre später waren es bereits mehr als 5.000 und zu Beginn der neunziger Jahre tummelten sich um die 300.000 Hosts im Netz [1].

So wünschenswert dieses Wachstum auch ist, so vielfältig sind die Probleme die dadurch entstehen. Als das Internet noch „klein“ und „ansehnlich“ war, wurden Lösungen eingesetzt, die sehr schnell an ihre Grenzen stießen. Ein anschauliches Beispiel dafür ist die hosts.txt, eine zentral verwaltete Datei, in der IP-Adressen einprägsame Namen zugeordnet werden (dazu mehr in Kapitel IV). In der Anfangszeit des Internets lud sich jeder Host regelmäßig diese hosts.txt von dem Standort, an dem sie gepflegt wurde. Bei ein paar hundert, vielleicht auch bei ein paar tausend Hosts ist das kein Problem. Jedoch ist auch ohne hellseherische Kräfte offensichtlich, dass bei einem solch schnellen Zuwachs an Hosts sehr bald eine Grenze erreicht wird, ab der eine solche Datei nicht mehr mit beherrschbarem Aufwand verwaltet werden kann.

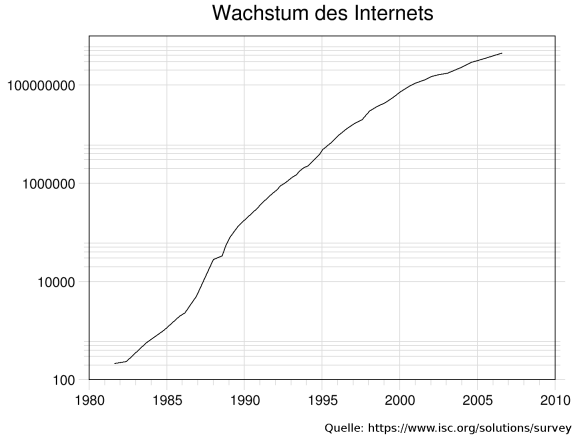

Abbildung 1. Anzahl der Hosts im Netz

Die folgenden Kapitel beschäftigen sich mit dem Wandel des Internets, der durch immer neue Notwendigkeiten stetig vorangetrieben wird. Auch wenn das Internetzeitalter mit Begriffen wie „Kurzlebigkeit“ oder „schneller Wandel“ assoziert wird, so stellt man doch fest, dass es an den Kernkomponenten des Netzes nur mit großer Vorsicht und sehr selten zu Veränderungen kommt.

II. NCP und TCP/IP

Das ARPAnet war das erste groß angelegte Paketvermittlungsnetzwerk und gilt als der Vorläufer des heutigen Internets [2]. Für die Umstellung der zu Grunde liegenden Architektur, damals NCP (=Network Control Program), auf TCP/IP, wurde am 1. Januar 1983, dem sogenannten flag-day, der Grundstein für das heutige Internet gelegt. Diese Umstellung war die vielleicht letzte große Revolution, die das Internet bis heute erlebt hat. Vermutlich war dieser Tag bis heute auch der letzte Tag, an dem ein solcher flag-day möglich war, denn damals war es „nur“ notwendig, etwa 400 Netzknoten auf die neuen Protokolle umzustellen. Bei der seit dem immer größer werdenden Anzahl an Rechnern müssten alle weiteren Änderungen verteilt werden, ohne das bis dahin bestehende Netzwerk zu kompromittieren [3].

A. Das Network Control Program

Sowohl für den Datentransport, als auch für die Vermittlung wurde im ARPAnet bis Ende 1982 das NCP (=Network Control Program) verwendet [4]. Zu diesem Zeitpunkt war die Anpassung oder Umstellung des Netzwerkprotokolls unumgänglich. Ein großes Problem bei NCP bestand darin, dass lediglich acht Bit für die Zieladresse eines jeden Pakets vergeben werden konnten. Damit war es nicht möglich, mehr als 256 Hosts direkt zu adressieren.

Die Anpassung der Länge der Zieladresse von NCP wäre eine Möglichkeit gewesen, um dieses Problem zu behebend. Stattdessen entschied man sich jedoch, die Transport- und die Vermittlungsschicht zu trennen und führte mit dem flag-day die Protokolle TCP (=Transmission Control Protocol)[1] und IP (=Internet Protocol)[2] ein.

Die letzte große Revolution war gelungen. Mittels TCP/IP wurde das Internet flexibler gestaltet und man hatte nun die Möglichkeit, Anpassungen an der Transport- bzw. Vermittlungsschicht vorzunehmen, ohne die jeweils andere Schicht zu beeinflussen.

B. Das Transmission Control Protocol

In seiner ersten Version war das TCP kaum mehr als ein Hilfsmittel zum Versenden von Daten, doch nun stand ein eigenständiges Protokoll auf der Transportschicht zur Verfügung, welches man verändern konnte, ohne die anderen Schichten zu kompromittieren. TCP war und ist sehr erfolgreich. Ein Großteil des Internettraffics wird über dieses Protokoll abgewickelt. Nichts desto trotz hat auch TCP Nachteile. Ein entscheidendes Problem ist der Overhead, welcher beim Versenden von Daten mittels TCP entsteht. Nicht nur der aufgeblähte Header (im Vergleich zu UDP), auch die Bestätigung (engl. Acknowledgement) des Empfängers zu jedem Paket und der Aufbau einer Verbindung zwischen Sender und Empfänger belasteten das Netz sehr stark. Schon Mitte der achtziger Jahre wurde die Überlastung des Netzes (engl. Congestion) zu einem nicht unerheblichen Problem, aufgrund dessen das Internet des öfteren kollabierte. Nach und nach wurde TCP um QoS-Eigenschaften und Congestion Control Algorithmen erweitert, um diesem Problem Herr zu werden (mehr zu Congestion Control mit TCP in Kapitel III).

C. Das Internet Protocol

Auch aus Sicht der Vermittlungsschicht hat sich die Revolution gelohnt. Das IP (damals Version 4, oder kurz IPv4) hat sich nach seiner Erscheinung nicht mehr geändert und es dauerte fast zehn Jahre, bis erste Probleme absehbar wurden. Durch die steigende Zahl an Rechnern im Internet und die viel zu großzügige Adressvergabe (siehe auch Kapitel VI) war es nur eine Frage der Zeit, bis der gesamte verfügbare IPv4-Adressraum aufgebraucht ist. Doch obwohl diese Problematik schon Anfang 1990 erkannt wurde und obwohl dies eine enorme Schwäche von IPv4 offenbarte, ist es bis heute nicht zu einer erneuten Revolution gekommen. Es wurde nachgebessert, bspw. kamen mit CIDR und NAT zwei neue Technologien hinzu, die das IP in seiner damaligen Form bis heute erhalten konnten.

Doch obwohl es bereits seit 1998 mit IPv6 [5] einen geeigneten Nachfolger gibt, ist es bisher nicht zu einer kompletten Umstellung gekommen. Im heutigen Internet fristet IPv6 noch immer ein Schattendasein und kommt nur selten zum Einsatz (dazu mehr im Kapitel VII).

III. Congestion Control

Aufgrund einiger Zusammenbrüche des Internets Mitte der achtziger Jahre, konnte auf die schnelle Einführung einer funktionierenden Congestion Control (zu deutsch: Überlastung- oder Staukontrolle) nicht mehr verzichtet werden [3]. Da die Überlastung des Netzes jedoch in erster Linie ein Problem der Vermittlungsschicht war[3], wäre eigentlich eine strukturelle Änderung an genau dieser Schicht oder zwischen Transport- und Vermittlungsschicht notwenig gewesen. Eine solche Änderung hätte jedoch einen sehr hohen Aufwand bedeutet.

Die einfachste Lösung war hingegen, TCP um einen Mechanismus zur Staukontrolle zu erweitern. Schließlich war TCP durch seine starke Verbreitung der Hauptverursacher von Netzüberlastungen.

A. Congestion Control in TCP

In TCP ist es ohne Weiteres möglich, Staus zu erkennen. Der Absender kann einfach davon ausgehen, dass sein Paket aufgrund eines Staus nicht angekommen ist, falls er nicht innerhalb einer bestimmten Zeit ein Acknowledgement (ACK) erhalten hat. Selbst wenn die Fehlerquelle kein Stau, sondern bspw. eine korrumpierte Routingtabelle eines Routers ist, so ist die Reduzierung der Netzlast immerhin kein schlechtes Verhalten.

Um einen Stau schnell aufzulösen, sollte jeder Sender, der einen Stau erkennt, mit Netzlastreduzierung reagieren. Dazu wurden die vier folgenden Algorithmen definiert [6]:

- Slow Start
- Congestion Avoidance
- Fast Retransmit
- Fast Recovery

Slow Start zusammen mit Congestion Avoidance

Um schon zu Beginn der Datenübermittlung einer möglichen Überlastung vorzubeugen, wird mit dem Slow Start Algorithmus das Congestion Window bestimmt. Da zu Beginn der Übertragung jedoch noch nichts über die Netzlast bekannt ist, wird anfänglich nur eine kleine Datenmenge verschickt. Der Empfänger antwortet auf jedes Datenpaket mit einem ACK, worauf der Sender selbst mit einer Vergrößerung des Congestion Window reagiert. Sobald die Pakete des Senders nicht mehr schnell genug mit ACKs bestätigt werden, ist die Slow Start Phase beendet.

[1] Zuständig für den Datentransport.
[2] Zuständig für die Datenvermittlung.
[3] Schließlich kann jedes beliebige Protokoll der Transportschicht Staus verursachen.

Danach tritt die Congestion Avoidance in Kraft. Dabei wird das Congestion Window nur noch dann vergrößert, falls alle Pakete aus dem Congestion Window mit einem ACK bestätigt wurden.

Kommt es im späteren Verlauf zu einem Timeout, setzt der Sender das Congestion Window wieder auf den Startwert und der Slow Start beginnt von Neuem. Diesmal wird die Slow Start Phase jedoch verkürzt, sodass das Congestion Window bei häufigen Paketverlusten nicht wieder zu schnell wächst.

Fast-Retransmit und Fast-Recovery

Es ist wünschenswert, dass nach Paketverlusten möglichst schnell auf Überlastungen reagiert wird. Zu diesem Zweck werden die Algorithmen Fast Retransmit sowie Fast Recovery verwendet. Hierzu muss der Empfänger dem Sender mitteilen, falls er Pakete in der falschen Reihenfolge erhält. Dies geschieht, indem Duplicate ACKs versendet werden. Eine Duplicate ACK ist eine Bestätigung des letzten korrekt empfangenen Pakets, für genau ein Paket welches außer der Reihe empfangen wurde.

Bemerkt der Sender Duplicate ACKs, beginnt er nach dem dritten Duplicate ACK sofort, das verlorene Paket erneut abzuschicken, ohne noch auf ein ACK dieses Pakets zu warten. Dies wird Fast Retransmit genannt, da nicht auf den Ablauf des Timers gewartet wird. Solche Duplicate ACKs sind ein Indiz dafür, dass es zwar zu einem Datenverlust gekommen ist, aber alle darauf folgenden Pakete den Empfänger erreicht haben. Deshalb wird das Congestion Window nach einem Fast Retransmit nicht auf den Startwert gesetzt, sondern nur halbiert.

Des Weiteren kann das Congestion Window um die Anzahl der Duplicate ACKs vergrößert werden, schließlich steht jedes Duplicate ACK für ein weiteres Paket, welches erfolgreich empfangen wurde. Durch dieses Prinzip kann der Sender nach einem Übertragungsfehler wieder schneller zur maximalen Übertagungsrate zurückkehren, weshalb man hier von Fast Recovery spricht.

Mit diesen vier Algorithmen hat man das Stauproblem vorübergehend in den Griff bekommen. Doch ist diese Evolution eher ein Notbehelf, als eine wahre Verbesserung der bestehenden Infrastruktur. Andere Protokolle, wie bspw. UDP werden in diesen Ansatz der Staukontrolle nicht mit einbezogen und können damit weiterhin das Netz überlasten. Da bis heute jedoch neben TCP kaum ein anderes Protokoll große Mengen an Netzlast verursacht hat, reichte diese Art der Staukontrolle bisher aus. Mit der immer stärkeren Nutzung von Video-Diensten, Onlinespielen und anderen UDP-basierten Anwendungen wird diese Evolution jedoch in absehbarer Zukunft an seine Grenzen stoßen.

IV. Der Domain Name Service

Die Einführung von DNS (=Domain Name Service) kann als eine echte Evolution angesehen werden. Die Nutzung einer eigenen Hosts-Datei blieb auch durch die Einführung des DNS möglich und am bereits bestehenden Teil des Internets mussten keine Änderung durchgeführt werden, die weitere Änderungen in anderen Teilen des Netzes nach sich gezogen hätten.

A. Die Hosts-Datei

Schon sehr früh wurde klar, dass die ursprünglich angedachte Lösung einer Hosts-Datei [7] zur Namensauflösung nicht ausreichen würde. Das exponentielle Wachstum des Internets machte die Idee einer zentral verwalteten, von Hand aktualsierten Liste, schnell obsolet.

Die Hosts-Datei würde schlicht und ergreifend zu groß werden und der Aufwand eine solche Datei auf dem aktuellen Stand zu halten, wäre innerhalb weniger Jahre mit unvorstellbarem Aufwand verbunden gewesen. War eine solche Liste in den Anfangszeiten des ARPAnets mit wenigen hundert Hosts noch praktikabel, so stellte sich bald heraus, dass eine andere Lösung gefunden werden musste.

Eine Möglichkeit wäre gewesen, ganz auf die Namensgebung zu verzichten, doch wer will und kann sich schon die Adresse 209.85.135.106 merken, um auf google.de zu gelangen? Auch hätte jeder Nutzer selbst eine Liste zur Benennung von IP-Adressen anlegen können. Doch wäre der Aufwand schon für wenige ausgewählte Seiten sehr hoch. Mit DNS hat sich jedoch eine bis heute praktizierte Lösung durchgesetzt.

B. Der Domain Name Service

DNS ist nichts Anderes, als ein weltweit verteilter Verzeichnisdienst, dessen ursprüngliche Aufgabe es war, die Hosts-Dateien durch eine effizientere Namensauflösung zu ersetzen. Durch den hierarchischen Aufbau (siehe Abb. 2) von DNS wurde dieses Problem gelöst und die Namensauflösung an unterschiedliche Administratoren deligiert.

In erster Linie wird DNS zur „Übersetzung“ von Domainnamen in IP-Adressen verwendet[4]. Gleiches gilt zwar auch für Hosts-Dateien, jedoch hat DNS den Vorteil, dass es linear skaliert und somit für das Internet deutlich besser geeignet ist. Zusammengefasst ergeben sich noch weitere Vorteile, hier die Wichtigstens im Überblick:

- Hierarchische Struktur (wodurch eine dezentrale Verwaltung umsetzbar ist)
- Eindeutigkeit
- Erweiterbarkeit

Sobald ein Client einen Namen auflösen möchte, stellt er eine Anfrage an einen DNS-Root-Server. Je nach Methodik, liefert dieser entweder den nächsten DNS-Server an den Client zurück (iterativ) oder er erfrägt beim nächsten DNS-Server die Adresse für den Client (rekursiv).

Durch spätere Erweiterungen des DNS konnten auch noch andere Anwendungen umgesetzt werden. So wurde es im Laufe der Zeit bspw. möglich, einem Domainnamen mehrere IP-Adressen zuzuweisen und somit per DNS die Last auf unterschiedliche Server zu verteilen[5] [8]. Diese und andere, später hinzugefügte Anwendungen wurden bei der Evolution der Hosts-Datei zu DNS jedoch noch nicht bedacht, weshalb hier nicht weiter auf sie eingegangen wird.

[4] Wobei der umgekehrte Weg auch möglich ist.
[5] Load Balancing

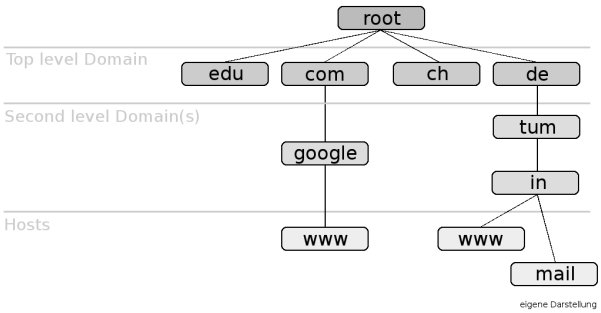

Abbildung 2. Teil des DNS-Namensraumes

V. BGP

Das BGP (=Border Gateway Protocol) wurde vor allem entwickelt, um sogenanntes policy routing zu ermöglichen. Dabei handelt es sich um die Fähigkeit eines Routing-Protokolls, verschiedene Regeln beim Routen von Paketen zu beachten.

Es wäre zum Beispiel denkbar, dass ein Paket von einem fremden System zu einem anderen fremden System geschickt werden soll und dieses Paket dabei das eigene autonome System durchlaufen will. Für zahlende Kunden wäre das durchaus in Ordnung. Nicht zahlende, fremde Parteien möchte man jedoch gerne daran hindern, da diese sonst das eigene System schnell überlasten könnten. Außerdem möchte man vielleicht Daten nur über Netzwerke anderer Parteien verschicken, falls es bei diesen zu einem geringen Preis möglich ist.

In der frühen Phase des Internets, als die Kommerzialisierung noch nicht weit vorangetrieben war, bestand hierfür keine Notwendigkeit. Durch Wachstum und Erweiterung auf immer neue Geschäftsfelder und Zielgruppen, wurde policy routing jedoch zwingend erforderlich. Erst dadurch war es wirtschaftlich sinnvoll, Anderen die eigene Netzinfrastruktur für deren Datenverkehr zur Verfügung zu stellen.

Heute wird BGP bereits in seiner vierten Version betrieben (kurz BGP4), welche ausschließlich zum Einsatz kommt. Der Vorteil von BGP4 gegenüber seinen Vorgängern ist die Möglichkeit, CIDR-Blöcken verarbeiten zu können (siehe Kaptiel VI).

VI. CIDR

Viele neue Technologien und Spezifikationen verdanken ihre Entwicklung und Nutzung vor allem dem rapiden Wachstum des Internets. Auch das Verfahren CIDR (=Classless Inter-Domain Routing) stellt hier keine Ausnahme dar. Anders als bei Classfull routing (mehr dazu in Abschnitt VI-A), werden bei CIDR die IPv4 Adressen weitaus weniger großzügig verteilt.

Durch eine frühzeitige Entwicklung und Einführung von IPv6 wäre die Entwicklung von CIDR für IPv4 nicht zwingend notwendig gewesen. Kurzfristig war es jedoch weniger Aufwendig, den bestehenden Adressraum besser aufzuteilen, als das alte Protokoll auf der Vermittlungsebene durch ein Neues zu ersetzen.

A. Class-A/B/C-Netze

Bis in das Jahr 1993 waren Class-A/B/C-Netze die verwendete Unterteilung des IPv4-Adressraums in kleinere Teilnetze. Da mit dieser Methodik der IPv4-Adressraum jedoch schon bald ausgeschöpft war, musste etwas passieren. Die Unterteilung in nur drei verschiedene Teilnetzgrößen war schlichtweg zu verschwenderisch, um dem rasanten Wachstum gerecht zu werden. Tabelle I zeigt, wie wenig Teilnetze nur mit Class-A -B und -C Netzen zur Verfügung stehen:

Tabelle I
ÜBERSICHT DER NETZKLASSEN

Netzklasse	Anzahl der Netze	Anzahl der Hosts pro Netz
Klasse A	128	16.777.214
Klasse B	16.384	65.534
Klasse C	2.097.152	254

Wie bereits erwähnt, wäre die wohl effektivste Lösung dieses Problems, der Umstieg auf einen größern Adressraum gewesen (siehe dazu Kapitel VII), denn selbst mit einer sparsameren Aufteilung der verfügbaren Adressen war abzusehen, dass ein Umstieg in nicht all zu ferner Zukunft notwendig wird. Eine solche Umstellung auf IPv6 ist jedoch nicht nur mit hohem Aufwand, sondern auch mit hohen Kosten verbunden, weshalb man sich für eine andere Lösung entschied.

B. Classless Inter-Domain Routing

CIDR teilt den Adressraum nicht in nur einige „wenige“ Teilnetze auf, sondern ermöglicht eine deutlich effizientere Nutzung des Adressraums. War es mit Classfull Routing notwendig, einer Organisation mit 400 Hosts ein Class-B-Netz zuzuteilen (und damit gut 65.000 IP-Adressen zu blockieren), so besteht mit CIDR die Möglichkeit, ein Teilnetz mit 512 IP-Adressen zu erzeugen. Damit wurde das Problem des zu kleinen Adressraums zwar nicht gelöst, aber zumindest in die Zukunft verschoben.

Neben diesem durchaus wichtigen Effekt, bot CIDR noch einen weiteren Vorteil. Für einen Router war es nun machbar, Routingtabellen deutlich zu verkleinert. Schließlich ist es mit CIDR häufig möglich, mehrere Adressen die über den gleichen Next Hop[6] erreicht werden, in einem Adressblock zusammenzufassen.

VII. INTERNET PROTOCOL VERSION 6

Die Umstellung von IPv4 nach IPv6 kommt nur schleppend voran. Hier handelt es sich um eine echte Evolution. Immer mehr Rechner und immer mehr Router werden für die neue Version des Internet Protocols „fit“ gemacht, weiterhin bleibt der Betrieb mit IPv4 möglich. Es gibt bereits einige Netzwerke, welche IPv6 für die Vermittlung verwenden und diese Netzwerke können ohne weiteres über 6to4-tunnel mit oder durch IPv4-Netzwerke kommunizieren. Dieser Parallelbetrieb ist jedoch aufwendig, was eigentlich zu einer raschen Umstellung führen sollte. Offensichtlich ist die Notwendigkeit jedoch

[6]Nächster bevorzugter Router, an dein ein IP-Paket weitergeschickt wird, falls es für den entsprechenden Adressbereich bestimmt ist.

noch nicht groß genug, um den Aufwand einer kompletten Umstellung zu rechtfertig.

Neben der wohl entscheidenden Änderung, endlich genug IP-Adressen zu Verfügung zu stellen[7], wurden bei der Entwicklung von IPv6 noch weitere wesentliche Ziele verfolgt:

- Verkleinerung von Routingtabllen
- Vereinfachung der Paket-Header, um Pakete von Routern schneller verarbeiten zu können
- Starke Unterstützung von Multicasting
- Die Möglichkeit für Host, ohne Adressänderung den Standort zu wechseln
- Sicherheitsfragen wurden berücksichtigt
- Abwärtskompatibilität

Obwohl, oder vielleicht gerade weil, IPv6 mit IPv4 über kleinere Umwege kompatibel ist, dauert die Umstellung schon sehr lange an und es ist heute noch nicht abzusehen, wann diese abgeschlossen sein wird. Da der Bedarf für IP-Adressen aber weiterhin stark steigt und große Teile der Erde bei der bisherigen Vergabe der Adressen nahezu leer ausgegangen sind, ist eine Umstellung auf lange Sicht unausweichlich.

VIII. Ausblick

Eins ist sicher, das Internet wird weiter wachsen und mit diesem Wachstum entstehen immer neue Anforderungen an Verfügbarkeit, Stabilität, Performanz, Kosten, Sicherheit,... weshalb immer wieder Änderungen an der bisherigen Architektur notwenig werden.

Doch in seiner jetzigen Form ist das Internet bereits viel zu groß, um eine Technologie komplett durch eine Andere zu ersetzen. Die Kosten wären nicht abschätzbar, die Kompatiblilität mit allen Teilnehmern im Netz könnte nicht gewährleistet werden und es ist nicht absehbar, ob eine neue Technologie in einem solchen Maßstab fehlerfrei funktioniert.

Für weitere Revolutionen, wie es sie das letzte Mal am 1. Januar 1983 gab, ist deshalb wohl kein Platz mehr. Mit Evolutionen kann man sich jedoch behelfen. Diese können die bereits funktionierende Infrastruktur nutzen, erweitern und verbessern, sowie alte Technologien - die ausgedient haben - nach und nach verdrängen.

Literatur

[1] "Internet systems consortium – www.isc.org/solutions/survey/history."
[2] James F. Kurose, Keith W. Ross, *Computer Networking – A Top-Down Approach*, 4th ed.
[3] M. Handley, "Why the internet only just works," *BT Technology Journal*, vol. 24, no. 3, pp. 119–129, Juli 2006.
[4] S. Crocker, "Protocol notes," RFC 36, March 1970.
[5] Andrew S. Tanenbaum, *Computernetzwerke*, 4th ed., 2003.
[6] M. Allman, "Tcp congestion control," RFC 2581, April 1999.
[7] M. Kudlick, "Host names on-line," RFC 608, Januar 1974.
[8] T. Brisco, "DNS support for load balancing," RFC 1794, April 1995.

[7]Würde man die Erde mit mit IPv6-Adressen bedecken, wäre auf jedem Quadratzentimeter der Erdoberfläche Platz für etwa 667 Billiarden Adressen.