

E-Mail - Elektronische Post *

Michael Dienert

8. Februar 2010

Inhaltsverzeichnis

1 Die beiden Hauptbestandteile des E-Mail-Systems	2
2 Transport von E-Mail mit dem Simple Mail Transport Protocol	2
2.1 smtp auf Schicht 4	3
2.2 Zeichensatz von E-Mail-Nachrichten	3
2.3 SMTP-Kommandowörter	3
2.4 Simulation einer SMTP-Sitzung mit einem Konsolenprogramm	3
3 Am Mail-Transport beteiligte Programme	4
4 Die E-Mail-Postfächer	5
5 Mailrouting	6
5.1 MX - Eintrag in einer DNS-Zonendatei (Resource Record)	6
6 Das Format einer E-Mail	6
7 E-Mail Anhänge	7
7.1 Codierung von Binärdaten mit 7-Bit-ASCII-Zeichen	9

E-Mail im Internet

Das Versenden von E-Mails ist eine der ältesten und beliebtesten Anwendungen des Internets. E-Mails gibt es, seit es Netzwerke in irgendeiner Form gibt.

Zu Beginn des Internets wurde eine E-Mail als Textdatei von einem Rechner zu einem anderen Rechner transportiert und dort im Postfach (mailbox, siehe Kap. 4) des Empfängers abgelegt. Und an diesem Prinzip hat sich bis heute nicht allzu viel geändert.

Durch das inzwischen sehr viel grösser gewordene Internet wurden allerdings einige Erweiterungen das Routing und die Verwaltung der Postfächer betreffend, notwendig.

* Auch wenn Ihr immer dachtet, emailiert werden nur Badewannen

Auch kann man inzwischen -maschinenunabhängig- Grafiken, Audiodateien, Filme, usw. als E-Mail-Anhänge übertragen. Leider gibt es für alle Belange keinen einzelnen, einheitlichen Standard. Vielmehr ist das E-Mail-System in einer Vielzahl von **RFCs** (Request For Comments) beschrieben.

1 Die beiden Hauptbestandteile des E-Mail-Systems

Die beiden wichtigsten Bestandteile des E-Mail-Systems sind in zwei RFCs, die bereits aus der Anfangszeit von E-Mail stammen festgelegt: **RFC 821**, verfasst 1982 von *Jonathan B. Postel*, beschreibt den **Transport** von E-Mail, die **RFC 822**, aus dem gleichen Jahr und von *David H. Crocker* geschrieben, beschreibt, wie eine E-Mail **aufgebaut** ist. Genauer gesagt, beschreibt RFC 822 wie der Kopfteil (Header) einer E-Mail auszusehen hat.

Aufbau und Transport einer E-Mail sind voneinander fast völlig *unabhängig*: In der RFC 821 wird z.B. nur an *einer* Stelle auf die RFC 822 verwiesen. Hier das Originalzitat:

```
mail data
```

```
A sequence of ASCII characters of arbitrary length,
which conforms to the standard set in the Standard
for the Format of ARPA Internet Text Messages
(RFC 822 [2]).
```

Und tatsächlich:

Es ist problemlos möglich, mit einem Mailserver nach RFC 821 eine E-Mail zu versenden, die überhaupt nicht RFC 822 entspricht.

Die Konsequenz hieraus ist: Wir können problemlos *alle* Header-Attribute einer E-Mail (s.u.) **fälschen**. Eine Tatsache, die sich vor allem Spammer zu Nutze machen, indem sie eine Fantasie-Absenderadresse angeben.

Inzwischen wurden die beiden Original-RFCs modernisiert und durch **RFC 2821** und **RFC 2822**, die beide 2001 erschienen, *abgelöst*.

RFC 2821 beschreibt nach wie vor den Transport von Mail. Es wurde jedoch um einige Dinge wie z.B. IPv6 und Unterstützung von Extended SMTP (ESMTP) erweitert, während veraltete Eigenschaften, wie z.B. die Unterstützung heute nicht mehr gebräuchlicher Layer3-Protokolle weggefallen ist.

Im Unterschied zu RFC 822 ist RFC 2882 ist deutlich genauer und strikter bei der Definition des Header-Aufbaus.

2 Transport von E-Mail mit dem Simple Mail Transport Protocol

Wie bereits erwähnt, ist der *Transport* von E-Mail durch RFC 821 festgelegt: Dort ist das *Simple Mail Transport Protocol (SMTP)* definiert.

SMTP ist ein Protokoll, das auf ISO-Schicht 7 arbeitet. Da es unabhängig von den darunter liegenden Schichten ist, kann man es mit verschiedenen Transport-Protokollen

wie z.B. TCP oder X.25 betreiben. Heutzutage wird aber ausschliesslich TCP im Internet benutzt, so dass es genügt, diese Kombination zu betrachten.

2.1 smtp auf Schicht 4

Unter TCP ist SMTP fest der **Port 25** zugeordnet. Der SMTP Dienst heisst `smtpd` (auf UNIX-Systemen). Da diese Programme für den Transport von Mail zuständig sind, werden sie auch **Mail Transfer Agent (MTA)** genannt.

2.2 Zeichensatz von E-Mail-Nachrichten

An dieser Stelle nun die wichtigste Eigenschaft (Unart) von SMTP:

SMTP arbeitet ausschliesslich mit **7-bit** Textzeichen aus dem **US-ASCII**-Zeichensatz. Das höchste Bit eines Bytes im TCP-Datenstrom wird deshalb immer auf '0' gesetzt.

Diese Einschränkung ist historisch bedingt. Sie ist heute absolut überflüssig und bedauerlich, da z.B. deutsche Umlaute mit 7-Bit ASCII Zeichen nicht dargestellt werden können.

2.3 SMTP-Kommandowörter

Wie der Name schon sagt, ist SMTP ein sehr einfaches Protokoll. Wie die RFC 821 vorschreibt, genügt es, dass ein SMTP-Dienst mindestens folgende Kommandowörter beherrscht:

HELO MAIL FROM: RCPT TO: DATA RSET NOOP QUIT

2.4 Simulation einer SMTP-Sitzung mit einem Konsolenprogramm

Um einen MTA zu überreden, eine Mail weiterzutransportieren, kann man sich mit `telnet` oder `nc (netcat)` über Port 25 mit einem `smtpd` verbinden und das vorgeschriebene Protokoll von Hand erzeugen.

Wie das im Detail vorgeht, zeigt folgende Kommando-Abfolge:

```
bash# nc localhost 25
<- 220 billie ESMTP Postfix
-> HELO billie
<- 250 billie
-> MAIL FROM: micha@localhost
<- 250 Ok
-> RCPT TO: micha@localhost
<- 250 Ok
-> DATA
<- 354 End data with <CR><LF>.<CR><LF>
-> das soll mal eine mehl geben.
-> .
<- 250 Ok: queued as E356D106E22
```

Wer das ausprobieren möchte: Die Doppelpunkte müssen wie gezeigt verwendet werden.

Wichtig ist, dass der DATA-Teil durch die Eingabe der Zeichenfolge

<CR><LF>.<CR><LF>

beendet wird. Die Abkürzungen bedeuten:

CR: Carriage Return (Wagenrücklauf), ASCII-Code 0x0A = 10

LF: Line Feed (Zeilenvorschub), ASCII-Code 0x0D = 13

Arbeitet man mit telnet oder nc wird <CR><LF> durch Betätigen der 'Enter'-Taste erzeugt.

Damit aber der SMTP-Server nicht durcheinanderkommt, wenn ein Benutzer beim Schreiben einer E-Mail mit einem E-Mail-Client ebenfalls diese Folge eingibt, wird jeder Punkt (.), der eingegeben wird und *am Anfang einer Zeile* steht *verdoppelt*. Der Mail-Client auf der Empfangsseite entfernt diese zusätzlichen Punkte wieder.

Hier nun der vom MTA (= Mail-Server) an die Datei /var/mail/micha (vgl. Kap. 4) angehängte Teil:

```
From micha@billie.billie Tue Mar 14 23:26:13 2006
Return-Path: <micha@billie.billie>
X-Original-To: micha@localhost
Delivered-To: micha@localhost.billie
Received: from billie (localhost [127.0.0.1])
        by billie (Postfix) with SMTP id E356D106E22
        for <micha@localhost>; Tue, 14 Mar 2006 23:21:28 +0100 (CET)
Message-Id: <20060314222129.E356D106E22@billie>
Date: Tue, 14 Mar 2006 23:21:29 +0100 (CET)
From: micha@billie.billie
To: undisclosed-recipients;

das soll mal eine mehl geben.
```

3 Am Mail-Transport beteiligte Programme

Die Abb. 1 zeigt eine Übersicht aller am Transport und Zustellung einer E-Mail beteiligten Programme.

Dabei bedeuten:

MUA: Mail User Agent oder auch *Mail-Client*. Das ist ein Anwender-Programm zum Versenden und Empfangen von E-Mail. Zum Versenden kontaktiert es einen SMTP-Server an Port 25 und zum Abholen der Mail verwendet es POP (aktuell POP3 an Port 110) oder IMAP (IMAP4 an Port 143). Bekannte MUAs sind: Pine (Kommandozeile, gut!), Outlook Express, Eudora, KMail, ...

MTA: Mail Transfer Agent oder auch *Mail-Server*. Der MTA erledigt den Transport der E-Mail. Verwendet wird dabei SMTP. Bekannte MTAs sind: Sendmail, Exchange, Postfix, Exim, QMail, ...

LDA: Local Delivery Agent. Der LDA nimmt E-Mail entgegen und verteilt diese auf die Postfächer. Ein LDA ist jedoch bei den meisten MTAs integriert.

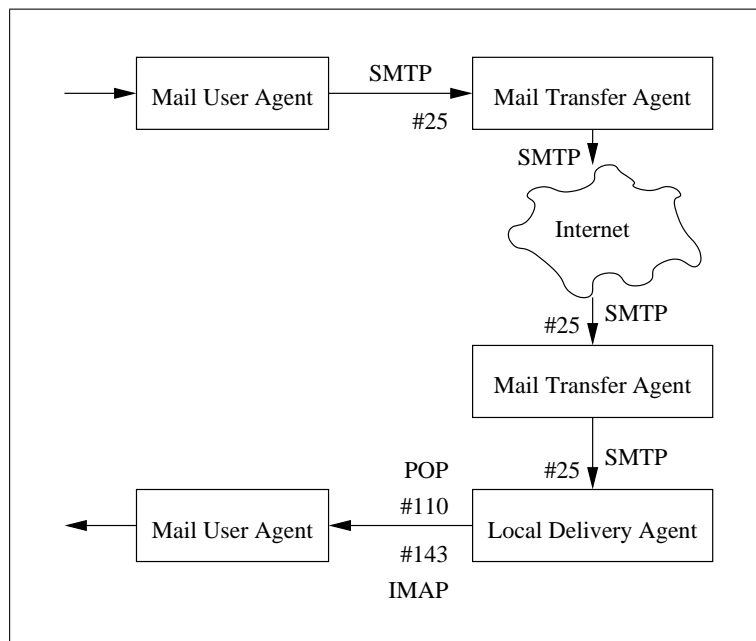


Abbildung 1: Komponenten beim Mail-Transport

4 Die E-Mail-Postfächer

Früher, als die IT-Welt noch schön und einfach war, waren die Hostrechner in einem Netzwerk fast ausschliesslich UNIX-Systeme.

Das oben erwähnte *Postfach* war und **ist** auf UNIX-Rechnern (dazu zählen natürlich auch Systeme mit GNU/Linux und die verschiedenen BSDs einschliesslich Apples OSX) eine **Textdatei**, die z.B. im Verzeichnis `/var/mail/` steht und den Benutzernamen des jeweiligen Benutzers hat.

Die Mailbox des Benutzers `alfred` heisst also `/var/mail/alfred`.

Da E-Mails nur aus 7-Bit-ASCII-Zeichen bestehen dürfen, kann man sie problemlos in einer lesbaren Textdatei speichern.

Trifft nun eine neue E-Mail für den Benutzer `alfred` ein, wird diese einfach an die Datei `/var/mail/alfred` **angehängt**.

Moderne Mail-Client-Programme prüfen regelmässig, ob sich die Mailbox-Datei geändert hat und zeigen die neu hinzugekommenen E-Mails an. Um den Inhalt der Mailbox zu holen hat man 2 Möglichkeiten:

1. Das Verzeichnis `/var/mail` wird auf allen Netzwerkrechnern mit NFS (Network File System¹) eingebunden und steht damit lokal zur Verfügung.
2. Das Mail-Client-Programm leert das Postfach mit dem *Post Office Protocol* (**POP3** an Port 110) oder dem *Internet Message Access Protocol* (**IMAP4** an Port 143)

¹Manche sagen auch Nightmare File System dazu

5 Mailrouting

Betrachtet man Bild 1 bleibt evtl. die Frage, wie denn eine E-Mail ihren Weg durch das nebulös dargestellte *Internet* findet.

Damit eine E-Mail richtig zugestellt werden kann, muss sie natürlich eine Empfängeradresse, eben die E-Mail-Adresse haben, die nach dem folgenden Schema aufgebaut ist:

`<name>@<maildomain>`

Hierbei bedeuten:

name: Ursprünglich war das der Unix-Benutzername des Empfängers (z.B. neumana). Heute kann das auch ein Name sein, mit dem der Benutzer eindeutig identifiziert ist (z.B. alfred.e.neumann) oder ein Alias-Name.

maildomain: Das kann

1. ein vollständiger DNS-Name (FQDN) des Mail-Servers (MTA) sein.
2. der Name einer DNS-Domäne sein, zu der der zuständige MTA gehört.

Ist nur ein Domain-Name angegeben, muss dieser noch in eine Hostadresse *aufgelöst* werden. Dazu muss es auf dem zuständigen DNS-Server einen Resource-Record mit dem Typ **MX** (Mail Exchange) geben, der den FQDN des MTA enthält. Über eine weitere Abfrage wird dieser dann endlich in eine IP-Adresse aufgelöst.

5.1 MX - Eintrag in einer DNS-Zonendatei (Resource Record)

Die Einträge in der Zonendatei eines DNS-Servers heissen *Resource Records*. Eine solcher Resource-Record für den Mail-Exchanger der Mail-Domäne `wara.de` sieht z.B. folgendermassen aus:

```
wara.de.    IN    MX    5    msv2.wrg.fr.bw.schule.de
wara.de.    IN    MX    10   msv1.wrg.fr.bw.schule.de
wara.de.    in    MX    15   xserve.wrg.fr.bw.schule.de
```

Hier sind nun 3 Mailexchange-Server angegeben. Durch die Zahl hinter **MX** wird die Priorität festgelegt, mit der die Mail-Server kontaktiert werden. Je kleiner der Wert, desto höher die Priorität. Auf diese Weise kann man leicht sicherstellen, dass auch bei Ausfall eines Mail-Servers weiterhin E-Mail zugestellt werden kann.

6 Das Format einer E-Mail

Wie bereits erwähnt, beschreibt RFC 2822 (ehemals RFC 822) den Aufbau und die Header-Schlüsselwörter einer E-Mail. Die eigentliche E-Mail ist für den MTA einfach nur eine Datei, die aus 7-Bit ASCII-Zeichen besteht. Diese Datei sollte jedoch dem in Abb. 2 gezeigten Aufbau entsprechen. Header und Body werden durch eine Leerzeile (CRLF) getrennt.

Der Body enthält den Text der E-Mail und kann auch in Bereiche unterteilt sein, die Anhänge enthalten. E-Mail-Anhänge sind im nächsten Kapitel beschrieben.

Der Mail-Header enthält eine Reihe von Schlüsselwörtern. Die allerwichtigsten sind hier aufgelistet:

FROM: Die E-Mail-Adresse des Absenders; diese kann einen völlig anderen Wert haben als der, den man im smtp-Protokoll nach FROM angibt!

SENDER: Name des Autors der E-Mail, falls dieser nicht identisch mit dem Benutzer ist, dessen Account verwendet wird.

TO: E-Mail-Adresse des Empfängers

REPLY-TO: E-Mail-Adresse an die eine evtl. Antwort gesendet werden soll

CC: Adressen von Empfängern, die eine Kopie der E-Mail erhalten sollen. CC ist die Abkürzung für Carbon Copy. Damit ist ein Durchschlag mit Kohlepapier (Carbon) gemeint.

BCC: Blind Carbon Copy. Wie CC, doch die Empfänger sehen nicht, wer noch eine Kopie erhalten hat.

SUBJECT: Der Betreff der Nachricht

X-SPAM-LEVEL: Headerfelder, die mit einem **X-** beginnen sind Erweiterungen, die nicht festgelegt sind. Es hängt nun vom MUA ab, wie und ob diese ausgewertet werden.

Eine einzelne Header-Zeile sieht allgemein so aus (der Doppelpunkt am Ende des Schlüsselworts ist wichtig):

Schlüsselwort: Wert

Zum Beispiel:

```
Subject: Das ist der Betreff
```

Ein Beispiel für eine vom MUA nicht aufbereitete E-Mail mit Header und Body steht oben (Kap. 2.4) .

7 E-Mail Anhänge

Da die RFCs 821 und 822 leider eine Menge Einschränkungen mitbringen, wurden in der Folge eine Reihe weiterer RFCs zu E-Mails verfasst.

Eingeführt wurden Mail-Anhänge (Attachments) mit der **RFC 1341** die beschreibt, wie der Datenteil einer E-Mail bei Bedarf in mehrere, voneinander unabhängige Bereiche aufgeteilt werden kann. Diese Bereiche dürfen Text- aber auch Nicht-Text-Daten wie z.B. Multimedia-Inhalte enthalten. RFC 1341 beschreibt somit eine "Erweiterung von E-Mail für vielfältige Zwecke": **Multipurpose Internet Mail Extension**, abgekürzt **MIME**.

Da das Versenden von Mail-Anhängen (sog. Attachments) beinahe zur Hauptaufgabe von E-Mails geworden ist, wurden hierzu zahllose weitere RFCs geschrieben. Heute aktuell ist eine ganze (!) Reihe: RFC 2045 - RFC 2049.

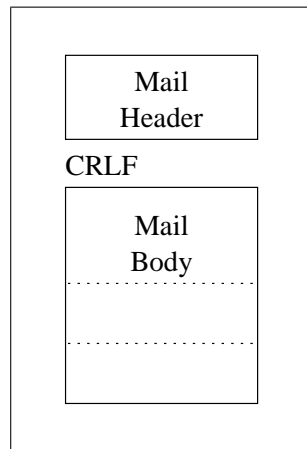


Abbildung 2: Struktur einer E-Mail

Der Typ des Inhalts (Text, Bilder, Audio, Binärdateien, ...) wird dem Empfänger in einem Header Feld durch den **MIME-Type** mitgeteilt. Dieses Feld heisst **Content-Type**.

Ein weiteres Header-Feld **-Content-Transfer-Encoding-** legt fest, wie die meistens 8-Bit breiten Binärdaten mit 7-Bit ASCII-Zeichen übertragen werden können (siehe unten).

Hier nun ein Beispiel für eine E-Mail mit einem jpg-Anhang. Das Header-Feld **Content-Disposition** wurde mit der RFC 1521 eingeführt und gibt den Dateinamen an, unter dem der Anhang gespeichert werden kann.


```

From: dienert@wara.de Tue Mar 21 00:20:10 2006
Mime-Version: 1.0 (Apple Message framework v623)
To: micha@localhost
Message-Id: <90d1c570e63a5964bf752caf31a13a4b@wara.de>
Content-Type: multipart/mixed;
    boundary=Apple-Mail-2--766630083
From: micha <dienert@wara.de>
Subject:
Date: Tue, 21 Mar 2006 00:20:10 +0100

--Apple-Mail-2--766630083
Content-Transfer-Encoding: 7bit
Content-Type: text/plain;
    charset=US-ASCII;
    format=flowed

eine i-mehl mit anhang (jpg)
--Apple-Mail-2--766630083
Content-Transfer-Encoding: base64
Content-Type: image/jpeg;
    x-unix-mode=0644;
    name="rennfahrerJo.jpg"
Content-Disposition: inline;
    filename=rennfahrerJo.jpg

/9j/4YpFRXhpZgAASUkqAAgAAAALAA4BAgALAAAAkgAAAA8BAgAGAAAAsgAAABABAgAGAAAyGAA
ABIBAwABAAAAQAABoBBQABAAAA2AAAABsBEQABAAAA4AAACgBAwABAAAAAgAAADEBAGAKAAA
6AAADIBAgAUAAAACAEAAABMCAwABAAAAgAAAGmHBAABAAAAHAEAAKQDAAAgICAgICAgICAgAAAA
AAAAAAAAAAAAAAAAAAAAAAAE5JS09OAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEU1NDAwAAAAAAAAAAAA

```

7.1 Codierung von Binärdaten mit 7-Bit-ASCII-Zeichen

Für die Übertragung von Binärdaten im Textteil einer E-Mail müssen die 8-Bit-Binärdaten in 7-Bit ASCII-Zeichen umcodiert werden.

Hierfür wurden eine Reihe von Codierungen entwickelt und mit dem Header-Feld Content-Transfer-Encoding wird dem Empfänger-MUA mitgeteilt, wie er wieder dekodieren muss.

Hier eine kleine Übersicht über gebräuchliche Kodierungen:

quoted-printable: Diese Kodierung dient nicht der Übertragung von Binärdateien sondern wird verwendet, wenn der Textteil einer E-Mail z.B. deutsche Umlaute enthält. Die jeweiligen Umlaute werden einfach durch Ihren Hexadezimalwert ersetzt, dem noch ein “=”-Zeichen vorangestellt wird:

- 'ä' wird zu '=E4'
- 'ö' wird zu '=F6'
- usw.

base64: Bei diesem Verfahren werden immer 3 Bytes der Binärdatei zu 24 Bit zusammengefasst. Diese 24-Bit-Sequenz wird dann wieder in 4 6-Bit-Wörter unterteilt. Diesen 6-Bit-Wörtern kann man dann leicht ASCII-Zeichen zuordnen. Ein Beispiel:

Binärfolge: 08 A0 FF (Hex) = 00001000 10100000 11111111

umgruppieren und ASCII-Zeichen zuordnen:

	000010	001010	000011	111111
Dezimalwert	2	10	3	63
ASCII-Zeichen	C	K	D	/

8bit: Der Mail-Textteil enthält dabei 8-Bit-Zeichen. Zur Zeit wird eine Weiterleitung solcher Mails nicht von allen MTAs unterstützt, es ist aber absehbar, dass das in Zukunft Standard werden wird.