

Inhaltsverzeichnis

1.0	Email und Messaging	2
1.1	Architektur und Dienste	2
1.2	Der User Agent	4
1.3	Nachrichtenformate	6
1.4	Nachrichtenübertragung	8
1.5	POP3	8
1.6	IMAP	9

Diese Dokumentation wurde verfasst von
Bernhard Kreinz
Systemspezialist und Webmaster

Email und Messaging

1.0 Email und Messaging

Layer 7	SMTP
Layer 4	TCP
Layer 3	IP

Stichworte:

Historische Email-Systeme
 ARPANET-Email und X.400
 User Agent und Message Transfer Agent
 Basisfunktionalitäten
 Mailboxen und Mailing-Listen
 Umschlag - Inhalt
 Header - Body
 Email-Reader
 MIME
 SMTP
 Email-Gateway
 POP3 und IMAP
 Spamming

1.1 Architektur und Dienste

Electronic Mail oder kürzer *Email* ist inzwischen recht verbreitet. Es ist doch immerhin seit fast zwei Jahrzehnten in Gebrauch. Die ersten Email-Systeme bestanden einfach aus Dateitransfer-Protokollen mit der Konvention, daß die erste Zeile der Nachricht (Message) die Empfängeradresse enthielt. Mit der Zeit wurden jedoch Schwachstellen dieser Technik offensichtlich:

Das Versenden von Nachrichten an eine Gruppe von Empfängern war sehr unbequem. Die Nachrichten hatten keine interne Struktur. So war beim Weiterleiten einer Nachricht nicht zu erkennen, was die originale Nachricht war und was durch die weiterleitenden Person hinzugefügte Informationen.

Der Absender wußte nie genau, ob die Nachricht angekommen war oder nicht

Email und Messaging

War jemand nicht an seinem Arbeitsplatz und wollte, daß alle eintreffenden Nachrichten von einem Stellvertreter bearbeitet werden, war dies schwer zu realisieren

Die Benutzerschnittstelle war zu roh. Zunächst mußte eine Datei in einem Editor erstellt werden und nach dem Abspeichern mit einem speziellen Transferprogramm abgeschickt werden.

Es war nicht möglich Nachrichten zu erzeugen, die aus einer Mixtur aus Text, Graphiken und gesprochener Sprache besteht

Aus diesen Gründen wurden mit der Zeit bessere Vorschläge für Email-Systeme gemacht. 1982 wurde der ARPANET Email-Vorschlag als RFC 821 (Übertragungsprotokoll) und RFC 822 (Nachrichtenformat) veröffentlicht. Diese wurden seither de facto Internet-Standards. Zwei Jahre später wurde der CCITT-Standard X.400 verabschiedet, der sich jedoch nie etablieren konnte.

Obwohl X.400 von einem offiziellen internationalen Standard, allen Telekombetrieben, vielen Regierungen und einem substantiellen Teil der Computer-Industrie unterstützt wurde, konnte eine handvoll Informatiker mit einem "Hack" ihr System weltweit durchsetzen. Dies lag weniger an der guten Qualität des ARPANET-Vorschlags als vielmehr an dem sehr schlechten und komplexen Design von X.400. Die Wahl zwischen einem einfachen und funktionierenden System gegenüber einem mächtigen aber nicht operablen System fiel der Netzwerkgemeinde nicht schwer.

Im folgenden soll ein Überblick über die Organisation von Email-Systemen gegeben werden. Sie bestehen normalerweise aus zwei Subsystemen: Der *User Agent*, der das Lesen und Verschicken von Email erlaubt und der *Message Transfer Agent*, der die Nachricht von Sender zu Empfänger bewegt. Der User Agent ist hierbei ein lokales Programm, das kommandobasierte, menübasierte oder graphische Methoden der Interaktion mit einem Email-System bereitstellt. Die Message Transfer Agents sind oftmals Hintergrundprozesse, die Email-Nachrichten durch das System transportieren.

Typischerweise unterstützen Email-Systeme fünf Basisfunktionalitäten, wie sie im folgenden beschrieben werden:

Die *Komposition* betrifft den Prozeß der Erzeugung von Nachrichten und Antworten. Obwohl ein einfacher Texteditor verwendet werden kann, bietet das System Hilfe bei der Erstellung der zahlreichen Email-spezifischen Informationsfeldern, die jede Nachricht beinhaltet.

Der *Transfer* referenziert auf die Bewegung der Nachricht von Sender zu Empfänger. Hauptsächlich erfordert dies den Aufbau einer Verbindung zu einem Zielrechner oder einem Zwischenrechner, das Abschicken der

Email und Messaging

Nachricht und das Lösen der Verbindung. Das Email-System sollte diese Arbeit automatisch durchführen.

Das *Reporting* hat die Aufgabe dem Sender mitzuteilen, was mit der Nachricht geschehen ist. Wurde sie ausgeliefert, wurde sie abgelehnt, ging sie verloren? In vielen Fällen ist eine Rückbestätigung der Auslieferung einer Nachricht essentiell.

Das *Anzeigen* erlaubt es den Benutzern, die eintreffenden Nachrichten zu lesen. Manchmal werden hierzu spezielle Konversionsschritte oder Präsentationskomponenten (z.B. für Graphiken) benötigt.

Die *Disposition* ist der letzte Schritt und betrifft die Aktionen, die ein Benutzer nach dem Empfang der Nachricht hat. Die Möglichkeiten reichen von Wegwerfen vor dem Lesen über Abspeichern bis hin zum Antworten. Weiterhin sollten Mechanismen zum Weiterleiten gegeben sein.

Die meisten Email-Systeme erlauben die Einrichtung von *Mailboxen* zum Abspeichern eintreffender Nachrichten. Hierzu werden Kommandos benötigt, die das Erzeugen und Löschen von Mailboxen, das Inspizieren des Inhalts sowie das Einfügen und Löschen einzelner Nachrichten betreffen.

Eine weitere Funktionalität, die oftmals gefordert wird, sind *Mailing-Listen*, die es erlauben eine Nachricht an eine ganze Gruppe von Benutzern zu schicken. Eine andere Option, die manchmal gewünscht wird, ist die *Registrierung* von Email. Damit erfährt der Absender, wann seine Nachricht beim Empfänger angekommen ist. Weitere Optionen sind Durchschläge, hoch priorisierte Email, sichere (verschlüsselte) Nachrichten, alternative Empfänger falls der erste nicht verfügbar ist und die Möglichkeit daß Sekretariate die Emails ihrer Chefs bearbeiten.

Ein Schlüsselkonzept moderner Email-Systeme ist die Unterscheidung zwischen dem *Umschlag* und dem *Inhalt*. Der Umschlag umschließt den Inhalt und enthält alle Informationen, um den Inhalt zu transportieren. Der Message Transport Agent nutzt diese Informationen für die Auslieferung der Nachricht. Die Nachricht in einem Umschlag besteht zumeist aus zwei Teilen, den *Kopf* (Header) und den *Körper* (Body). Der Header enthält dabei Kontrollinformationen für den User Agent. Der Body ist ausschließlich für den menschlichen Benutzer gedacht.

1.2 Der User Agent

Der User Agent - auch *Email Reader* genannt - akzeptiert normalerweise eine Reihe von Kommandos für die Komposition, dem Empfang und die

Email und Messaging

Beantwortung von Nachrichten sowie für die Manipulation von Mailboxen. Einige User Agents haben aufwendige Menü- oder Graphik-basierte Benutzerschnittstellen für die man eine Maus benötigt, während andere einfache Tastatureingaben erwarten. Funktional gesehen sind sie meist identisch.

Startet man einen Email Reader, so wird er typischerweise zuerst in die Eingangs-Mailbox (Incoming) des Benutzers schauen ob für ihn eine Nachricht angekommen ist. Dann wird er die Anzahl der eingetroffenen Nachrichten oder einzeilige Zusammenfassungen aller Nachrichten anzeigen. Weitere Angaben können dabei zusätzlich auf dem Bildschirm erscheinen: Die Nachricht ist neu, die Nachricht ist nicht neu wurde aber in der Mailbox belassen, auf die Nachricht wurde schon geantwortet, die Nachricht wurde an jemanden "geforwarded", usw. Weitere Angaben können die Größe der Nachricht in Bytes sein, der Absender und ein sogenanntes *Subject*. Das Subject-Feld ist eine Analogie zu der Betreff-Zeile in einem Brief und wird in Email-Readern zumeist als Zusammenfassung angezeigt. Daher ist es in Emails recht wichtig ein aussagekräftiges Subject zu formulieren.

Eine Sammlung häufig implementierter Kommandos auf einem Email Reader zeigt die folgende Liste:

Anzeigen der Subjects einer Mailbox

Anzeigen der ersten Zeilen mehrerer Nachrichten als Liste

Senden einer Nachricht Forwarden einer Nachricht

Eine Nachricht beantworten

Eine Nachricht löschen

Eine Undelete-Operation auf gelöschte Nachrichten anwenden

Eine Nachricht in einer anderen Mailbox speichern

Die Nachrichten nach Beendigung des Mail Readers in der Eingangs-Mailbox behalten

Eine andere Mailbox lesen

Zur nächsten oder vorhergehenden Nachricht gehen und sie anzeigen

Zu einer speziellen Nachricht gehen und sie anzeigen

Um eine Email abzuschicken, muß der Benutzer zunächst die Nachricht formulieren, die Zieladresse angeben und möglicherweise weitere Parameter zur Verfügung stellen. Insbesondere die Zieladresse muß in einem Format angegeben werden, die der User Agent versteht. Viele User Agents akzeptieren DNS-Adressen in der Form *mailbox@location*, z.B. "bk@barnes.ch". Aber auch die X.400-Adressen lohnen einen Blick. Sie

Email und Messaging

sehen viel anders aus als die DNS-Adressen. Sie werden aus *Attribut = Wert*-Paaren gebildet:

/C=US/SP=MASSACHUSETTS/L=CAMBRIDGE/PA=360 MEMORIAL
DR./CN=KEN SMITH/

Diese Adresse spezifiziert Land, Staat, Ort, persönliche Adresse und einen Namen. Viele andere Attribute sind möglich. Daher kann man eine Email an jemanden schicken, dessen Namen man nicht kennt, über den man aber viele andere Informationen besitzt. Typischerweise würde man jedoch zumeist *Aliase* (benutzerspezifische Kurzbezeichnungen) verwenden, auch bei den DNS-Adressen.

1.3 Nachrichtenformate

Wie schon weiter oben gesagt bestehen Emails aus einem primitiven Umschlag (RFC 821), einigen Header-Feldern, einer leeren Zeile zur Abtrennung und einem Nachrichtenkörper. Jedes Header-Feld besteht aus einem ASCII-String, der einen Feldnamen, einen Doppelpunkt und einem Wert besteht. Die grundlegenden Header-Felder wie sie im RFC 822 definiert sind, sind in der nachstehenden Tabelle aufgelistet:

Header	Bedeutung
To:	Email-Adressen der primären Empfänger
Cc:	Email-Adressen weiterer Empfänger
Bcc:	Email-Adressen für Durchschläge
From:	Person, die die Nachricht erstellte
Sender:	Email-Adresse des Absenders
Received:	Eine Zeile wird hier von jedem Transfer Agent auf dem Weg zum Ziel hinzugefügt
Return-Path:	Kann genutzt werden um einen Weg zurück zum Absender zu identifizieren

Tabelle 5.1: RFC 822 Header-Felder

Zusätzlich zu den RFC 822 Feldern können noch eine Reihe von anderen Feldern genutzt werden. Die gebräuchlichsten listet die folgende Tabelle auf:

Header	Bedeutung
Date:	Datum und Zeit wann die Nachricht abgeschickt wurde (mit Zeitzone)
Reply-To:	Email-Adresse an wen Antworten geschickt werden sollen

Email und Messaging

Message-Id:	Einzigartige Nummer, mit der die Nachricht später referenziert werden kann
In-Reply-To:	Message-Id der Nachricht zu der diese Nachricht eine Antwort ist
Reference:	Andere relevante Message-Ids
Keywords:	Benutzerspezifische Schlüsselworte
Subject:	Kurze Zusammenfassung der Nachricht (Betreff-Zeile)

Tabelle 5.2: Zusätzlich Header-Felder

Nach dem Header kommt der Nachrichtenkörper. Ein Benutzer kann dort jedes gültige Zeichen hineinschreiben. Einige Leute beenden ihre Nachrichten mit einer Signatur mit Namen und Adresse, kleinen ASCII-Cartoons und Anmerkungen von sehr unterschiedlicher Phantasie und Wichtigkeit.

In den frühen Tagen von ARPANET beinhalteten Emails ausschließlich textuelle Nachrichten in Englisch und im ASCII-Standard. Heute beinhalten Nachrichten spezielle Buchstaben verschiedener Sprachen, z.B. Umlaute oder Buchstaben mit Akzenten Buchstaben aus einem nicht-lateinischen Alphabet, z.B. Hebräisch oder Russisch Elemente aus Sprachen ohne Alphabet, z.B. Chinesisch oder Japanisch nichttextuelle Elemente, z.B. Bilder, Video oder Audio

Als Lösung für solche Nachrichten wurde in RFC 1341 und RFC 1521 die *Multipurpose Internet Mail Extension* (MIME) vorgeschlagen, das inzwischen weltweit genutzt wird. Grundsätzlich wird bei MIME eine Struktur zum Message-Körper hinzugefügt, die die Kodierungsregeln für Nicht-ASCII-Nachrichten definiert. Alles was dann noch getan werden muß, ist das absendende und das empfangende Programm zu ändern. Diese MIME-fähigen Programme kodieren/dekodieren alle Nicht-ASCII-Elemente in einer speziellen Art, daß sie auf ASCII-Basis übertragen werden können.

MIME-Elemente werden mit einem speziellen Header gekennzeichnet. Die gängigsten MIME-Typen zeigt die folgende Tabelle auf.

Typ	Untertyp	Beschreibung
Text	Plain	Unformatierter Text
Text	Richtext	Text mit einfachen Formatierungskommandos
Bild	Gif	Bild im GIF-Format
Bild	Jpeg	Bild im JPEG-Format
Audio	Basic	Wiedergabefähiger Sound

Email und Messaging

Video	Mpeg	Video im MPEG-Format
Applikation	Octet-stream	Eine uninterpretierte Byte-Sequenz
Applikation	Postscript	Ein druckbares Dokument im Postscript-Format

Tabelle 5.3: MIME-Typen und -Untertypen wie sie in RFC 1521 definiert werden.

1.4 Nachrichtenübertragung

Im Internet wird eine Email von einer Quelle verschickt, indem die zugehörige Maschine eine TCP-Verbindung auf Port 25 zu einer Zielmaschine etabliert. An diesem Port lauscht auf der Empfängerseite ein Hintergrundprozeß (Daemon), der das *Simple Mail Transfer Protocol* (SMTP) versteht. Der Daemon akzeptiert eintreffende Verbindungen und kopiert Nachrichten zu ihren zugehörigen Mailboxen. Kann eine Nachricht nicht ausgeliefert werden, wird eine Fehlermeldung zurückgeschickt. SMTP ist ein einfaches ASCII-Protokoll. Nach Aufbau der TCP-Verbindung über Port 25 agiert die sendende Maschine als Client und die empfangende Maschine als Server. Die Größe der Nachricht sollte jedoch in vielen Fällen 64 kBytes nicht übersteigen. Der Server schickt die Nachricht mit Hilfe von DNS weiter, bis der eigentliche Empfänger erreicht ist.

Email über SMTP funktioniert am besten, wenn sich sowohl Sender als auch Empfänger auf dem Internet befinden und TCP-Verbindungen aufnehmen können. Jedoch sind nicht alle Maschinen, die Email schicken oder empfangen wollen, auf dem Internet. In diesem Falle existieren *Email Gateways*, die von einer Email-Konvention zur anderen übersetzen können, z.B. von SMTP nach X.400.

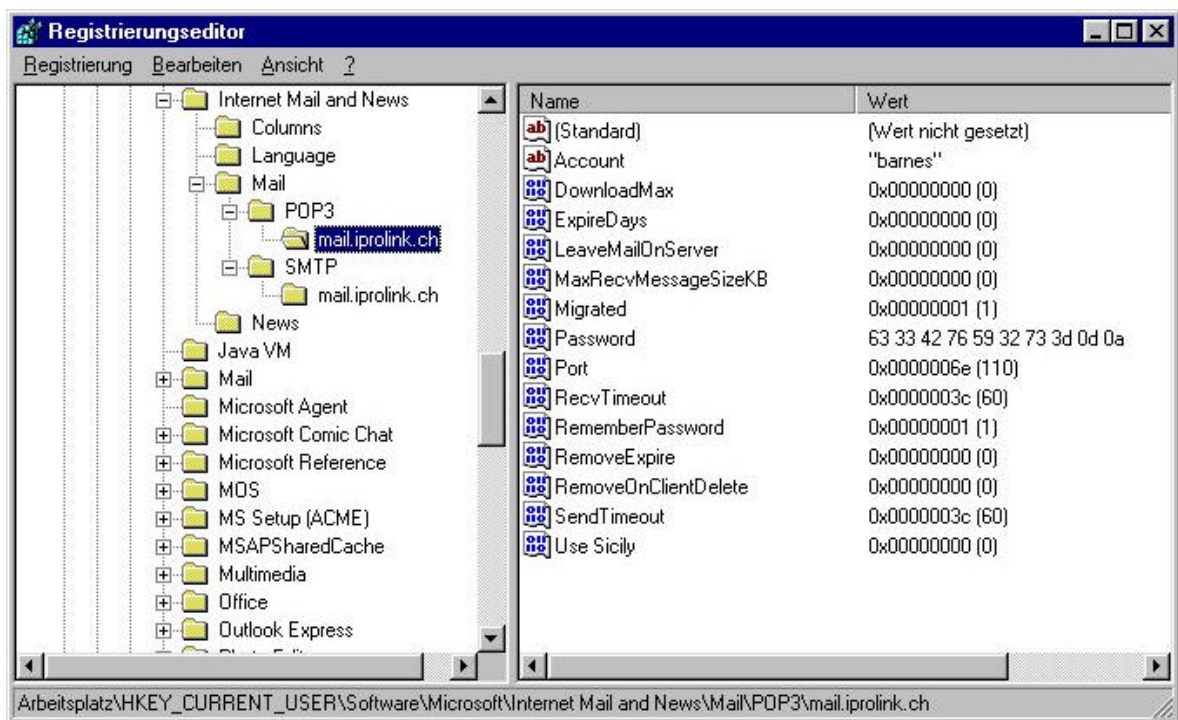
Bisher sind wir von der Situation ausgegangen, daß eine Maschine direkt aus dem Internet Email empfangen kann. Jedoch ist es in vielen Firmen die Email-Infrastruktur auf der Basis eines zentralen Email-Servers aufgebaut. Die Benutzer holen sich auf diesem Server mit Hilfe von Client-Rechnern ihre Emails ab.

1.5 POP3

Layer 7	POP3
Layer 4	TCP
Layer 3	IP

Email und Messaging

Das gebräuchlichste Protokoll hierfür ist das *Post Office Protocol 3* (POP3), wie es in RFC 1225 definiert ist. Eine Client-Software mit diesem Protokoll hat Kommandos um sich beim Server einzuloggen, Nachrichten abzuholen und Nachrichten zu löschen. Das Protokoll selbst sieht dabei ähnlich wie SMTP aus. Die grundlegende Idee bei POP3 ist, daß alle Email-Nachrichten vom Server auf den Client kopiert werden und die lokalen Nachrichten dort verarbeitet werden.



(Beispiel einer POP3 Konfiguration anhand Win95/NT)

1.6 IMAP

Layer 7	IMAP
Layer 4	TCP
Layer 3	IP

Ein etwas weitergehendes und moderneres Protokoll ist das *Interactive Mail Access Protocol* (IMAP), das in RFC 1064 definiert wird. Es wurde entwickelt, um mehrere Client-Rechner zu unterstützen zwischen denen

Email und Messaging

sich der Benutzer bewegt. Daher ist die Grundidee von IMAP, die Email-Nachrichten immer an einem zentralen Ort vorzuhalten.

Spamming

Kaum ein Thema sorgt in letzter Zeit für soviel Gesprächsstoff in der Internet-Gemeinde wie die steigende Zahl unerwünschter Emails. "Werden Sie Millionär in 30 Tagen" oder "Bei uns bekommen Sie den besten Rechner des Universums" sind Botschaften, die zuhauf in den Mailboxen eintreffen.

Dieses sogenannte *Spamming* - benannt nach einem amerikanischen Büchsenfleisch zweifelhafter Qualität - geht inzwischen über das Ärgernis hinaus. Firmen verlieren beträchtliche Geldsummen durch Spamming-Angriffe auf den Email-Server. Internet-Provider müssen immer öfter Kundenbeschwerden wegen Spam-Mails bearbeiten. So sind dies beispielsweise bei Earthlink, Pasadena inzwischen 1000 bis 3000 Beschwerden pro Tag!

Bei einer Umfrage im April 1997 sagten über 15% der Benutzer, daß sie sehr viel Spam-Mail erhalten, ca. 24% viel, ca. 28% wenig, ca. 20% wenige und nur ca. 12% keine. War vor noch nicht allzu langer Zeit die Antwort der Internet-Gemeinde auf Spamming das sogenannte *Mail-Bombing*, d.h. das Überfluten des Absenders mit Antworten, so wird dies immer schwieriger. Kommerzielle Spam-Versender nutzen spezielle Internet-Technologien, um ihre Absenderadresse zu verbergen.

Bei zu grosser Belästigung empfiehlt es sich, mal kurz beim Provider nachzufragen, ob diese Email gefiltert werden kann, ansonsten muss der Email Client benutzt werden ...