Inhaltsverzeichnis

So funktioniert E-Mail

Aus: http://www.tecchannel.de/kommunikation/e-mail/401772/index.html

Über ein halbe Milliarde Menschen besitzen ein E-Mail-Postfach. Doch wie kommt eine E-Mail an ihr Ziel? Wir erläutern die grundlegende Technik, den Aufbau einer E-Mail und die Protokolle **SMTP**, **POP** sowie **IMAP**.

Trotz neuer Alternativen wie Instant Messenging wird die E-Mail immer beliebter. Das Marktforschungsunternehmen IDC prognostiziert für das Jahr 2005 rund 36 Milliarden versandte E-Mails pro Tag. Im Jahr 2000 verfügten rund 505 Millionen Menschen weltweit über ein E-Mail-Postfach, im Jahr 2005 soll es bereits 1,2 Milliarden elektronische Briefkästen geben.

Dabei hatte die E-Mail im Herbst 1971 einen eher unspektakulären Start. Der BBN-Techniker Ray Tomlinson versandte eine Mail zwischen zwei Rechnern, die über das damalige Arpanet miteinander verbunden waren. Auf der Suche nach einem unverbrauchten Satzzeichen für die elektronische Post entdeckte er dabei das @-Zeichen und definierte so das Symbol für ein neues Zeitalter.

Einen weiteren Meilenstein in der Geschichte der elektronischen Post legte Eric Allman mit der Programmierung der Software **Sendmail** im Jahr 1981. Damit war es erstmals möglich, Nachrichten mit einem Mailprogramm gleichzeitig in verschiedene Netze zu versenden.

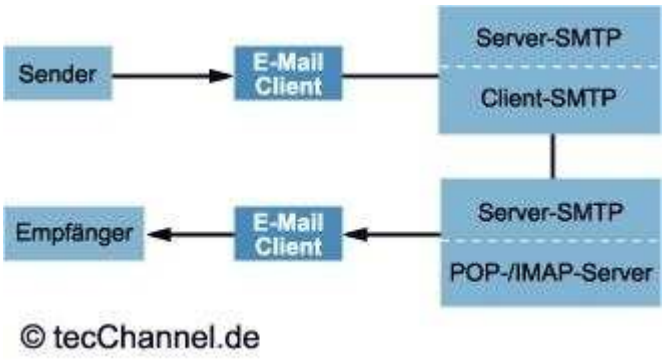
Der heutige Erfolg der E-Mail war 1971 freilich noch nicht absehbar, Tomlinsons Erfindung machte nur wenige Schlagzeilen. Heutzutage ist die elektronische Post kaum mehr wegzudenken und gehört für viele zum Alltag.

Die E-Mail-Kommunikation basiert auf drei Protokollen: **SMTP zum Versenden** und **POP und IMAP zum Empfangen** von Nachrichten. Die Spezifikationen für jedes Protokoll sind jeweils in einem oder mehreren RFCs festgelegt.

1 SMTP - Simple Mail Transfer Protocol

Die Aufgabe des Simple Mail Transfer Protocol (SMTP) ist der zuverlässige und effiziente Transport von Nachrichten. SMTP ist unabhängig vom Netzprotokoll, in der Regel wird das im Internet übliche TCP verwendet. Die Kommunikation erfolgt über den **Port 25**.

Für den Austausch von Nachrichten sind so genannte **Mail Transfer Agents (MTA)** zuständig. Der bekannteste MTA ist Sendmail. Anwender kommen normalerweise mit diesen nicht in Kontakt. E-Mail-Clients wie Outlook und KMail übernehmen die Übertragung der elektronischen Post von und zum Mail Transfer Agent. Die MTAs verwenden zur Kommunikation untereinander einfache ASCII-Zeichen (Standard: **7-Bit**-ASCII). Der Client sendet Kommandos zum Server, der mit einem nummerischen Code und einem optionalen String antwortet.



SMTP: Die Grafik erläutert den Weg einer E-Mail vom Sender zum Empfänger.

Das Simple Mail Transfer Protocol hat jedoch einen grossen Nachteil: Nach dem Versenden einer E-Mail erhält man keine weiteren Informationen über deren Verbleib. Die Spezifikationen setzen zwar eine Benachrichtigung des Versenders voraus, falls eine Mail nicht zugestellt werden kann. Wie eine solche auszusehen hat, wurde nicht festgelegt. Meist ist dies eine Mail mit einer Fehlermeldung und dem angehängten Header der unzustellbaren Nachricht. Auf Grund eines fehlenden Standards lässt sich in der Praxis nur selten herausfinden, wo und warum Fehler aufgetreten sind.

Daher wurde eine neue **SMTP-Erweiterung** für standardisierte Fehlermeldungen ins Leben gerufen. Allerdings unterstützen derzeit nur wenige Server die Erweiterung, so dass diese hier nicht näher behandelt wird. Interessierte finden in den RFCs 1891 und 1894 weitere Informationen.

1.1 Kommandos

Die SMTP-Kommandos definieren den Mailtransport. Laut Spezifikation muss eine Implementation von SMTP mindestens folgende acht Kommandos unterstützen:

Wichtige SMTP-Kommandos

Kommando	Beschreibung
EHLO oder HELO	Extended HELLO oder HELLO: Startet eine Sitzung und identifiziert den Client am Server. Als Argument wird, sofern verfügbar, der Fully Qualified Domain Name (FQDN) des Client übergeben. Ansonsten sollte eine andere Kennung zur Identifizierung gesendet werden, beispielsweise der Rechnername.
MAIL	Startet eine Mailübertragung. Als Argument wird die Absenderadresse (reverse-path) übergeben.
RCPT	Recipient: Identifiziert den Empfänger (forward-path) einer Mail. Bei mehreren Empfängern wird das Kommando mehrmals ausgeführt.
DATA	Der Server antwortet auf das Kommando mit dem Code 354 und wartet auf die Übertragung der Nachricht. Der Client beendet die Übertragung mit "CRLF"."CRLF".
RSET	Reset: Die Mailtransaktion wird abgebrochen. Die Verbindung zwischen beiden Rechnern bleibt

Kommando	Beschreibung
	jedoch bestehen.
VRFY	Verify: Überprüft eine Empfänger-Adresse.
EXPN	Expand: Die meisten MTAs wie Sendmail behandeln das Kommando wie VRFY.
NOOP	Bewirkt die Antwort "250 OK" vom Server. Dient zur Aufrechterhaltung der Verbindung, ohne dass es einen Time-Out gibt.
QUIT	Beendet die Verbindung. Der Server muss daraufhin die Antwort "250 OK" zurückliefern.

1.2 Antwortcodes

Die SMTP-Antwortcodes garantieren, dass der Client jederzeit über den Status des Servers informiert ist. Jedes Kommando erfordert einen Antwortcode vom Server. Der Client entscheidet ausschliesslich anhand des zurückgelieferten nummerischen Codes über das weitere Vorgehen.

SMTP-Antwortcodes

Code	Beschreibung
211	System-Status oder System-Hilfe.
214	Hilfe - Informationen zum Ausführen eines Kommandos.
220	Server bereit.
221	Server beendet Verbindung.
250	Kommando ausgeführt.
251	Keine lokale Mailbox; Weiterleitung an "forward-path".
252	Überprüfung der Empfängeradresse nicht möglich; Die Nachricht wird dennoch versendet.
354	Starte Empfang der Mail; Beenden mit "CRLF". "CRLF".
421	Service nicht verfügbar; Verbindung wird beendet.
450	Aktion nicht ausgeführt - Mailbox nicht verfügbar.
451	Aktion abgebrochen - Fehler beim Ausführen.
452	Aktion abgebrochen - Nicht genügend System-Speicher.
500	Syntax-Fehler - Kommando unbekannt.
501	Syntax-Fehler - Parameter oder Argument falsch.

Code Beschreibung

502 Kommando unbekannt / nicht implementiert.

503 Falsche Reihenfolge der Kommandos.

504 Parameter unbekannt / nicht implementiert.

550 Aktion nicht ausgeführt - Mailbox nicht erreichbar (nicht gefunden, kein Zugriff).

551 Mailbox nicht lokal; "forward-path" versuchen.

552 Aktion abgebrochen - Fehler bei der Speicherzuweisung.

553 Aktion nicht ausgeführt - Mailbox-Name nicht erlaubt (Syntax inkorrekt).

554 Transaktion fehlgeschlagen (beim Verbindungsaufbau: Kein SMTP-Service verfügbar).

1.3 Envelope, Header und Body

Eine E-Mail besteht aus drei Teilen:

- Envelope: Beinhaltet den Sender und Empfänger einer Nachricht und wird von den Mail Transfer Agents benötigt.
- Header: Verwendet der Mail-Client für weitere Informationen wie Client-Kennung und Message-ID.
- Body: Enthält den eigentlichen Text der Nachricht. RFC822 spezifiziert den Body als ASCII-Text.

Beim Versenden einer Mail mit dem Kommando DATA überträgt der Client den Header, gefolgt von einer Leerzeile und dem Body. Jede übertragende Zeile darf nicht länger als 1000 Bytes sein.

Hier sehen Sie ein Beispiel für einen Header:

```
Received: by xyz.de. id AA00502; Mon, 19 Nov 2001 12:47:32 +0100
Received: from adam1 (715684625313-0001@[192.168.80.201]) by fwd00.xyz.de
with smtp id 166Cyz-1KXYRsC; Tue, 20 Nov 2001 16:38:45 +0100
From: adam@xyz.de (Adam)
To: eva@test.de (Eva)
Subject: Beispiel-Mail
Date: Mon, 19 Nov 2001 12:47:31 +0100
Reply-To: adam@xyz.de
Message-ID: <9307191947AA00502.Adam@xyz.de>
MIME-Version: 1.0
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: 8bit
X-Mailer: Microsoft Outlook IMO, Build 9.0.2416 (9.0.2910.0)
```

Unter *Received* werden alle SMTP-Server eingetragen, die die E-Mail auf dem Weg vom Sender zum Empfänger passiert hat. Jede Nachricht erhält eine eindeutige Kennung, die *Message-ID*. Meist besteht diese aus einer Zahlen-/Buchstaben-Kombination, gefolgt von der Host-Adresse des Senders. Mit X-beginnende Zeilen sind in der Regel vom Mail-Client hinzugefügte Informationen, die für den Versand der Nachricht nicht zwingend erforderlich sind. Die Zeilen *Mime-Version*, *Content-Type* und *Content-Transfer-Encoding* kennzeichnen eine MIME-konforme Mail. Alle weiteren Zeilen wie *Date* oder *Subject* sind weit gehend selbst erklärend.

1.4 Beispiel für den Versand einer Mail

In einem Beispiel wird eine dreizeilige Nachricht an zwei Empfänger versendet:

```
S: 220 test.de SMTP server ready
C: HELO xyz.de.
S: 250 xyz.de., pleased to meet you
C: MAIL From:<adam@xyz.de>
S: 250 <adam@xyz.de> Sender ok
C: RCPT To:<eva@test.de>
S: 250 <eva@test.de> Recipient ok
C: RCPT TO:<tom@test.de>
S: 250 <tom@test.de> Recipient ok
C: DATA
S: 354 Enter mail
C: Hallo Eva, hallo Tom!
C: Beispiel für den Mail-Versand mit SMTP.
C: Adam
C: .
S: 250 Mail accepted
C: QUIT
S: 221 test.de delivering mail
```

Zum Versenden einer Nachricht sind fünf Kommandos notwendig: Nachdem der Mail-Client über TCP eine Verbindung zum SMTP-Server aufgebaut hat, wartet er auf einen Begrüssungstext mit dem Antwortcode 220. Im nächsten Schritt identifiziert sich der Client mit dem Kommando HELO, als Argument übergibt er den Fully Qualified Domain Name seines Host, in diesem Beispiel *xyz.de*. Das Kommando MAIL identifiziert den Verfasser der Nachricht. Das Kommando RCPT gibt die Empfänger an. Den Inhalt einer Mail sendet der Client mit dem Befehl DATA. Das Ende der Nachricht kennzeichnet eine Zeile, die nur einen Punkt enthält. QUIT beendet die Verbindung, und der Server versendet die Nachricht.

1.5 Mail Routing und das DNS

Nachdem der SMTP-Server eine Nachricht vom Client entgegengenommen hat, ist er für das Routing der E-Mail verantwortlich.

Das **Domain Name System** spielt nicht nur beim Zugriff auf Web- oder FTP-Server eine zentrale Rolle, sondern auch beim Versand elektronischer Nachrichten. Für E-Mails sind im DNS spezielle Einträge vorgesehen: die **MX-Records**.

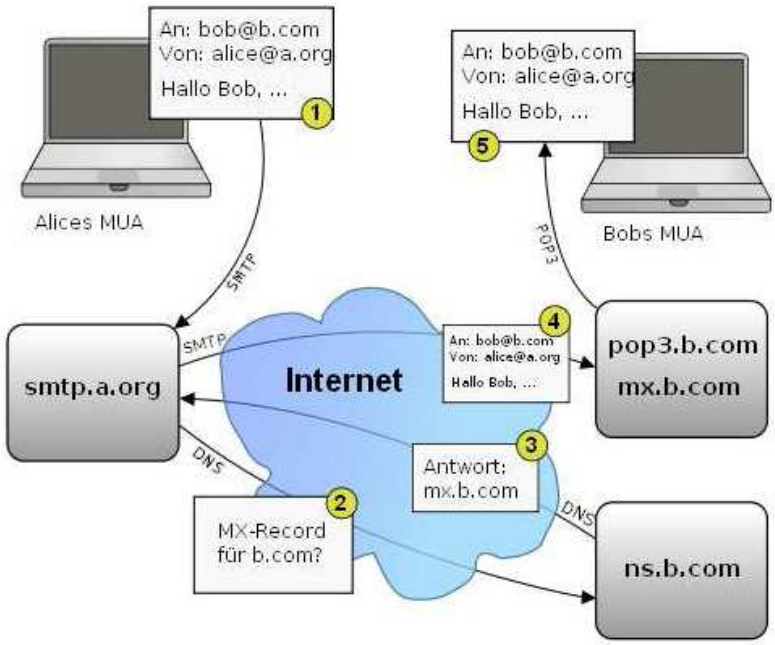
Bsp. **DNS-Eintrag** für **gmx.ch**:

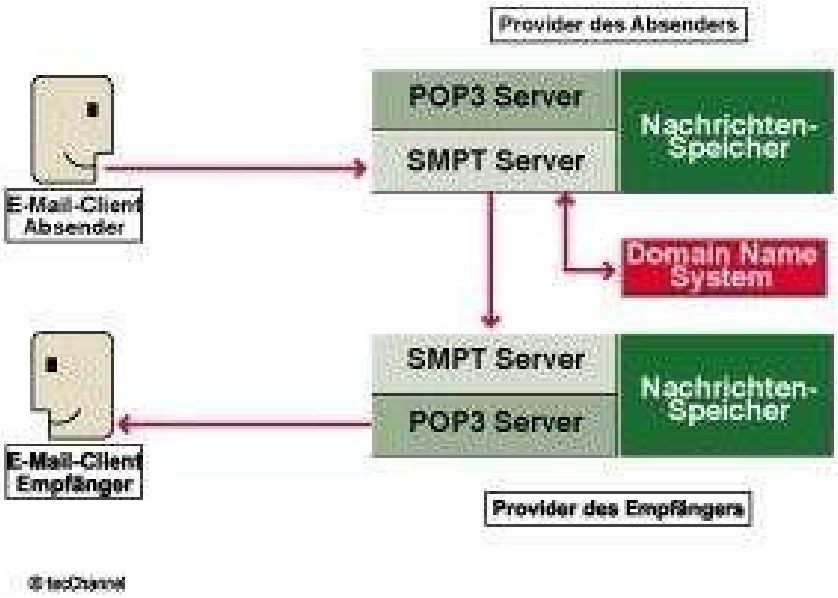
```
gmx.ch.  1D IN SOA        dns.gmx.net. hostmaster.gmx.net. (
                          2008060300  ; serial
                          8H                          ; refresh
                          2H                          ; retry
                          1W                          ; expiry
                          1H )                        ; minimum

gmx.ch.  1D IN TXT        "v=spf1 redirect=gmx.net"
gmx.ch.  1D IN MX 10      mx0.gmx.de.
gmx.ch.  1D IN MX 10      mx0.gmx.net.
gmx.ch.  1D IN A          213.165.65.50
gmx.ch.  1D IN NS         ns.schlund.de.
gmx.ch.  1D IN NS         dns.gmx.net.
```

Der Server identifiziert den Zielrechner über den so genannten **Mail Exchange Record (MX Record)** der Domain. Dazu fragt er einen DNS-Server ab und erhält eine Liste mit Servern (Mail Exchanger), die Nachrichten für die Ziel-Domain entgegennehmen. Jeder Mail Exchanger ist mit einer 16-Bit langen Priorität versehen. Der SMTP-Server versucht nun, in der Reihenfolge der Priorität dem entsprechenden Server die Nachricht zu übermitteln.

Symbolischer Ablauf:





DNS und E-Mail: Nach der Übergabe einer Mail an den SMTP-Server konsultiert dieser das DNS, um die Empfängerseite zu identifizieren.

Prinzipiell kann eine Nachricht über mehrere SMTP-Server laufen. Entgegen der weit verbreiteten Meinung, E-Mails könnten mehrere Male um den Globus wandern, bis sie schliesslich beim Empfänger eintreffen, überqueren sie **meist nur zwei SMTP-Server**. MX Records sollen das Entstehen von "Mailschleifen" verhindern.

Dennoch kann es in Ausnahmefällen zu solchen Mailschleifen kommen. Dies ist beispielsweise der Fall, wenn Routing-Informationen unvollständig oder nicht mehr aktuell sind. Beim Umzug einer Domain zu einem anderen Provider tritt dies eventuell auf.

1.6 Extended SMTP

Im Laufe der Jahre sind die **Anforderungen an die E-Mail-Kommunikation gestiegen**. Um dieser Entwicklung Rechnung zu tragen, wurde SMTP um einige Kommandos und Funktionen erweitert. Diese Erweiterungen hat man im Protokoll ESMTP (Extended SMTP) festgelegt. Alle neu hinzugekommenen Funktionen sind abwärtskompatibel, bereits existierende Implementationen sind somit nicht betroffen.

Nutzt ein Client die erweiterten Features, identifiziert er sich beim SMTP-Server mit dem Kommando EHLO (statt HELO). Ist der Server zu den Erweiterungen kompatibel, antwortet er mit einem mehrzeiligen Antwortcode 250. Jede Zeile enthält ein Kennwort und ein optionales Argument. Die Kennwörter spezifizieren die SMTP-Erweiterungen, die der Server unterstützt.

```
220 test.de SMTP server ready
EHLO xyz.de
250-xyz.de, pleased to meet you
```

```
250-HELP
250-EXPN
250-8BITMIME
250-SIZE 461544960
250 XADR
```

Der Antwortcode und das Kennwort werden durch einen Bindestrich getrennt, ausgenommen die letzte Zeile, die ein Leerzeichen enthält. Die Kommandos HELP und EXPN gibt es zwar bereits seit der ersten SMTP-Spezifikation, da diese aber optional ist, werden sie bei ESMTP oft zusätzlich angegeben. Alle Kennwörter, die mit einem "X" beginnen, verweisen auf lokale SMTP-Erweiterungen.

Wichtige SMTP Extensions

Extension	Beschreibung
8BITMIME	Erlaubt das Verwenden von **8-Bit-ASCII-Zeichen** im Body (Standard: 7-Bit-ASCII); Spezifiziert in RFC1426.
SIZE	Gibt die maximale Grösse einer Mail an (in Bytes), die der Server akzeptiert; Spezifiziert in RFC1427.

1.7 Multipurpose Internet Mail Extension

Wie bereits erwähnt verwendet man zum Senden von E-Mails im Body **7-Bit-ASCII-Text**. Dieser umfasst nur 128 Zeichen, internationale Sonderzeichen kommen darin nicht vor. Die unter anderem in Deutschland gebräuchlichen Umlaute wären somit in elektronischen Nachrichten nicht verwendbar. RFC2045 definiert MIME (Multipurpose Internet Mail Extension), das die Probleme beseitigt, wenn in E-Mails andere Zeichensätze als US-ASCII Verwendung finden.

Der Body einer MIME-Mail kann weiterhin als ASCII-Text übertragen werden, ohne Rücksicht auf dessen Inhalt. Einzige Voraussetzung für den Einsatz ist die Unterstützung durch den E-Mail-Client. MIME fügt dem Header einige Elemente hinzu, die dem Empfänger die Strukturierung des Bodys erläutern:

MIME-Header:

Element	Parameter	Beschreibung	Beispiel
MIME-Version	1.0	Kennzeichnet die verwendete MIME-Version. Derzeit existiert nur Version 1.0.	MIME-Version: 1.0
Content-Type	text, image etc.; Es folgt nach einem "/" der Subtyp.	Bestimmt den Inhalt der Mail. Bei den Typen "text" und "multipart" wird eine Zeichensatzangabe und Textkörper-Kennung ergänzt.	Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-	7bit, 8bit, binary, quoted-printable	Kennzeichnet den Algorithmus, in dem die Daten vorliegen.	Content-Transfer-Encoding: 8bit

Element	Parameter	Beschreibung	Beispiel
Encoding			

1.8 MIME-Typen

MIME ermöglicht nicht nur die Übertragung von E-Mails mit Anhängen, sondern stellt auch gleichzeitig die Informationen zur Verfügung, die ein E-Mail-Client zur Auswahl des Darstellungsprogramms benötigt. Dazu dient im Header das Element "Content-Type". MIME unterteilt die Datentypen (Media-Types) in sieben Obergruppen mit mehreren Untergruppen. Jede Dateiendung wird einem "Media-Type" zugewiesen.

Wichtige MIME-Typen

Typ	Typ	Beschreibung
text	plain	Unformatierter Text.
	richtext	Text mit einfachen Formatierungen, zum Beispiel Kursiv.
	enriched	Vereinfachte und erweiterte Form von richtext.
multipart	mixed	Mehrere Body-Teile, die sequenziell bearbeitet werden.
	parallel	Mehrere Body-Teile, die parallel bearbeitet werden.
	digest	Auszug aus einer E-Mail.
	alternative	Mehrere Body-Teile mit identischem logischen Inhalt.
message	rfc822	Inhalt ist eine andere RFC822-Nachricht
	partial	Inhalt ist das Fragment einer E-Mail.
	external-body	Inhalt ist ein Zeiger zur eigentlichen Nachricht.
application	octet-stream	Binär-Daten.
	postscript	PostScript-Daten.
image	jpeg	ISO10918-Format.
	gif	CompuServe Graphic Interchange Format (GIF).
audio	basic	Kodiertes 8-Bit-µ-law Format.
video	mpeg	ISO11172-Format

2 Empfangen von Mails

Viele Anwender nutzen das Internet über eine Wählverbindung oder lassen ihren Computer nicht dauernd laufen. Diese können daher auf ihren Rechnern nicht ohne Weiteres einen SMTP-Server zum Empfang von Mails einrichten. Dazu müsste der SMTP-Server des Providers die Nachrichten an den eigenen Server weiterleiten, was in den meisten Fällen nicht möglich ist. Aus diesem Grund werden die Nachrichten zum Abruf zwischengespeichert. Für den Fernzugriff auf diese E-Mail-Verzeichnisse sind **zwei Protokolle von Bedeutung**: Das Ältere ist **POP**, das seit 1984 zur Verfügung steht. Die aktuelle Version ist **POP3**. Der zweite Ansatz ist das 1986 entwickelte **IMAP**, das in der erweiterten Version 4rev1 vorliegt.

In einer verteilten E-Mail-Struktur gibt es laut RFC1733 drei Möglichkeiten zum Zugriff auf Nachrichten:

- Bei der Offline-Verarbeitung werden Mails an einen Server übermittelt. Ein Client greift auf den Server zu und lädt die Nachrichten herunter. Die Bearbeitung der elektronischen Post erfolgt lokal auf dem Rechner des Benutzers.
- Im Online-Betrieb bleibt die Mail auf dem Server und wird dort vom Client bearbeitet.
- Die dritte Möglichkeit ist der Disconnected-Zugriff, ein Hybrid aus Online- und Offline-Modell. Bei dieser Variante nimmt der Client Verbindung zum Server auf, erstellt eine Kopie der Nachrichten und baut die Verbindung wieder ab. Nachdem der Benutzer die Mail bearbeitet hat, erfolgt ein Abgleich der Mails zwischen Client und Server.

2.1 POP versus IMAP

IMAP bietet im Vergleich zu POP zahlreiche Vorteile, besonders für den Fernzugriff auf E-Mails. Die folgende Tabelle bietet einen Überblick über den Funktionsumfang beider Protokolle.

IMAP und POP im Überblick

Funktion	IMAP	POP
Unterstützung für den Offline-Modus	X	X
Unterstützung für den Online-Modus	X	
Unterstützung für den Disconnected-Modus	X	
Mails laufen beim SMTP-Server auf	X	X
Für den Empfang von Nachrichten ist kein SMTP-Gateway erforderlich, jedoch für den Versand	X	X
Zugang zu unterschiedlichen Mailboxen möglich	X	
Erlaubt Zugang zu anderen Daten, wie beispielsweise News	X	
Hohe Performance bei Verbindungen über Leitungen mit niedriger Bandbreite (Modems)	X	
Message-Status-Flags lassen sich bearbeiten	X	
Neue Nachrichten sind mit unterschiedlichen Clients überall im Netz zugänglich	X	X
Offene Protokolle; Spezifiziert in RFCs	X	X
Zusatzprotokoll für die Verwaltung von Benutzereinstellungen verfügbar (Internet Message Support Protocol, IMSP)	X	

Auf den folgenden Seiten finden Sie eine detaillierte Erläuterung der Protokolle POP3 und IMAP4.

2.2 POP3: Post Office Protocol, Version 3

Wenn ein Client über POP3 Nachrichten abrufen möchte, baut er eine TCP-Verbindung über **Port 110** auf. Ist die Verbindung zustande gekommen, sendet der Server eine Begrüssungsmeldung. Die weitere Kommunikation zwischen beiden Rechnern erfolgt über Kommandos.

POP3-Kommandos bestehen aus drei oder vier Zeichen langen Kennwörtern und einem oder mehreren Argumenten mit bis zu je 40 Zeichen. Antworten enthalten einen Status-Indikator und ein Kennwort sowie optionale Informationen. Es gibt zwei Status-Indikatoren: Positiv (+OK) und Negativ (-ERR).

Eine POP3-Verbindung durchläuft mehrere Sitzungsstufen. Nachdem der Server seine Begrüssungsmeldung gesendet hat, beginnt der "Authorization State". Der Client muss sich gegenüber dem Server identifizieren. Verläuft dies erfolgreich, beginnt der "Transaction State". Es werden alle Operationen zum Bearbeiten von Mails, wie Löschen und Abrufen der Nachrichten, ausgeführt. Sendet der Client das Kommando QUIT, beginnt der "Update State". Der Server beendet die TCP-Verbindung und führt die vom Client in den "Transaction State" angeforderten Änderungen durch.

Viele POP3-Server haben zusätzlich einen Inaktivitäts-Timer. Laut Spezifikation muss dieser auf mindestens zehn Minuten eingestellt sein. Jedes Kommando seitens des Client setzt den Timer zurück. Ist der Timer abgelaufen, wird die TCP-Verbindung sofort beendet, ohne in den "Update State" zu wechseln - eventuelle Änderungen werden auf dem Server nicht gespeichert.

2.2.1 Authorization State

Wenn der POP3-Client eine TCP-Verbindung zum Server aufgebaut hat, sendet dieser eine einzeilige Begrüssungsmeldung. Dies kann jeder beliebige String sein:

```
S: +OK POP3 server ready
```

Dabei handelt es sich bereits um eine Antwort des Servers, daher beginnt die Begrüssungsmeldung immer mit einer positiven Bestätigung (+OK). Die Verbindung befindet sich nun im "Authorization State". Der Client muss sich jetzt gegenüber dem Server identifizieren. Dies erfolgt über die beiden Kommandos USER und PASS.

Kommandos im "Authorization State"

Kommando	Argument	Beschreibung
USER	Name	Das Argument identifiziert eine Mailbox.
PASS	String	Der String enthält ein Mailbox-spezifisches Passwort.
QUIT	-	Beendet die Verbindung.

2.2.2 Transaction State

Hat sich der Client Server identifiziert, wechselt die Verbindung in den "Transaction State". Dem Client stehen nun eine Reihe von Kommandos zur Behandlung der Mails zur Verfügung:

Kommandos im "Transaction State"

Kommando	Argument	Beschreibung
STAT	-	Liefert die Anzahl der gespeicherten Mails und die Grösse der Mailbox in Oktetts zurück.
LIST	Nummer	Liefert die Nummer und Grösse (in Oktetts) aller Mails zurück. Wird als Argument eine Mail-Nummer angegeben, wird nur die Grösse dieser Mail ausgegeben.
RETR	Nummer	Gibt die Mail mit der als Argument übergebenen Nummer aus.
DELE	Nummer	Löscht die Mail mit der übergebenen Nummer.
NOOP	-	Bewirkt die Antwort "+OK". Dient zur Aufrechterhaltung der Verbindung, ohne dass es einen Time-Out gibt.
RSET	-	Setzt die aktive Verbindung zurück. Noch nicht ausgeführte Änderungen werden verworfen.

Der Server führt das Kommando DELE nicht unmittelbar aus. Die entsprechenden E-Mails werden zum Löschen gekennzeichnet und erst bei Beenden der Verbindung endgültig vom Server gelöscht. Hat man eine Nachricht zum Löschen gekennzeichnet, möchte dies jedoch rückgängig machen, führt man das Kommando RSET aus. Der Server verwirft alle noch nicht ausgeführten Operationen.

2.2.3 Update State

Wenn der Client das Kommando QUIT sendet, wechselt die Verbindung in den "Update State". Der Server trennt die TCP-Verbindung und führt alle gespeicherten Änderungen aus.

Kommando im "Update State"

Kommando	Argument	Beschreibung
QUIT	-	Beendet die TCP-Verbindung und führt alle gespeicherten Änderungen aus.

Neben den hier vorgestellten, für eine minimale Implementation erforderlichen Kommandos gibt es noch einige weitere, die von den meisten Clients und Servern unterstützt werden. Details hierzu finden Sie in RFC1725.

2.2.4 Beispiel für das Abrufen einer Mail

In diesem Beispiel sehen Sie den Ablauf einer POP3-Verbindung. Der Client identifiziert sich gegenüber dem Server und ruft eine Liste der gespeicherten E-Mails ab. Danach werden die Nachrichten einzeln heruntergeladen, auf dem Server zum Löschen gekennzeichnet, und die Verbindung wird beendet.

```
S: +OK POP3 server ready
C: user tecchannel
S: +OK
C: pass ahd635d
S: +OK
C: LIST
S: +OK 2 messages (320 octets)
S: 1 120
S: 2 200
S: .
C: RETR 1
S: +OK 120 octets
S: <Server sendet Nachricht 1>
S: .
C: DELE 1
S: +OK message 1 deleted
C: RETR 2
S: +OK 200 octets
S: <Server sendet Nachricht 2>
S: .
C: DELE 2
S: +OK message 2 deleted
C: RETR 3
S: -ERR no such message
C: QUIT
S: +OK
```

2.3 IMAP4: Internet Message Access Protocol, Version 4

E-Mail-Client und Server tauschen bei IMAP ihre Daten über den **TCP-Port 143** aus. Im Gegensatz zu den Protokollen SMTP und POP muss der Client bei IMAP nicht nach jedem gesendeten Kommando auf die unmittelbare Antwort des Servers warten. Es können mehrere Befehle hintereinander versendet werden, die jeweilige Rückmeldung vom Server kann später erfolgen. Dazu wird jedem Kommando seitens des Client eine Kennung vorangestellt, auch "Tag" genannt, zum Beispiel "A001" für den ersten Befehl und "A002" für den zweiten. Der Server kann dem Client auf mehrere Arten antworten: Mit einem Plus-Zeichen am Anfang der Zeile antwortet der Server, wenn er weitere Informationen zu dem vorangegangenen Kommando erwartet. Er signalisiert dem Client gleichzeitig seine Empfangsbereitschaft. Steht dagegen ein Sternchen am Anfang der Zeile, sendet der Server weitere Informationen an den Client zurück.

© tecChannel.de

Sitzungsstufen: Eine IMAP4-Verbindung durchläuft mehrere Phasen.

Die Antwort eines Servers kennzeichnet den Erfolg oder Fehler eines Kommandos: "OK" (Kommando erfolgreich ausgeführt), NO (Fehler beim Ausführen) oder BAD (Protokoll-Fehler: Kommando unbekannt oder Syntax-Fehler). Die Antwort enthält denselben Tag wie das zugehörige Kommando, so erkennt der Client, welcher Response welchem Befehl gilt. Wie bei POP durchläuft eine IMAP-Verbindung mehrere Sitzungsstufen:

- Non-Authenticated State: Unmittelbar nach dem Aufbau der Verbindung. Der Anwender muss sich gegenüber dem Server identifizieren.
- Authenticated State: Der Anwender hat sich erfolgreich identifiziert und muss nun eine Mailbox auswählen.
- Selected State: Eine Mailbox wurde ausgewählt. In dieser Phase lassen sich die Mailbox und Mails bearbeiten.
- Logout State: Die Verbindung wird abgebrochen, und der Server führt noch anstehende Änderungen aus.

Auf den folgenden Seiten werden die einzelnen Sitzungsstufen detailliert erläutert.

2.3.1 Non-Authenticated State

Der "Non-Authenticated State" stellt mehrere Möglichkeiten zur Identifizierung des Anwenders zur Verfügung. Folgende Kommandos stehen in diesem Verbindungs-zustand bereit:

Kommandos im "Non-Authenticated State"

Kommando	Argument	Beschreibung
AUTHENTICATE	Authentifizierungs-Mechanismus	Das Kommando bestimmt den Authentifizierungs-Mechanismus, zum Beispiel "Kerberos" oder "S/Key". Details zu den Authentifizierungs-Mechanismen finden Sie in RFC1731.
LOGIN	Name / Passwort	Identifiziert den Anwender über Benutzername und Passwort.

Beispiel für eine Authentifizierung mittels des Kommandos LOGIN:

```
C: a001 LOGIN EVA AHD635D
S: a001 OK LOGIN completed
```

2.3.2 Authenticated State

Im "Authenticated State" hat sich der User authentifiziert und muss nun eine Mailbox auswählen, welche in dieser Sitzung bearbeitet werden soll. Dazu stehen unter anderem folgende Kommandos zur Verfügung:

Wichtige Kommandos im "Authenticated State"

Kommando	Argument	Beschreibung
SELECT	Mailbox-Name	Wählt eine Mailbox zur weiteren Bearbeitung aus. Als erfolgreiche Antwort sendet der Client Informationen zur gewählten Mailbox, wie beispielweise die Anzahl der gespeicherten Nachrichten.

Kommando	Argument	Beschreibung
EXAMINE	Mailbox-Name	Identisch mit dem Kommando SELECT. Jedoch wird die Mailbox als "read-only" ausgewählt, es sind keine dauerhaften Änderungen möglich.
CREATE	Mailbox-Name	Erstellt eine Mailbox mit dem als Argument übergebenen Namen.
DELETE	Mailbox-Name	Löscht die als Argument übergebene Mailbox.
RENAME	Bestehender Mailbox-Name / Neuer Mailbox-Name	Ändert den Namen einer Mailbox.

Beispiel zum Löschen einer Mailbox mit DELETE:

```
C: A683 DELETE FRIENDS
S: A683 OK DELETE completed
```

2.3.3 Selected State und Update State

Im "Selected State" stehen zahlreiche Kommandos zum Bearbeiten einer Mailbox zur Verfügung:

Wichtige Kommandos im "Selected State"

Kommando	Argument	Beschreibung
CLOSE	-	Entfernt alle zum Löschen gekennzeichneten Mails und setzt die Verbindung in den Authenticated State zurück.
EXPUNGE	-	Entfernt alle zum Löschen gekennzeichneten Mails, die Verbindung bleibt im Selected State.
SEARCH	ein oder mehrere Suchkriterien	Erlaubt die Suche nach bestimmten Nachrichten in der aktuellen Mailbox. Das Kommando unterstützt Boolesche Verknüpfungen.
FETCH	Gewünschte Daten einer Nachricht	Bewirkt das Senden von Daten einer Nachricht vom Server zum Client.

Beispiel zum Suchen einer bestimmten Nachricht mit SEARCH. Als Ergebnis der Suche liefert der Server die Nummern der entsprechenden Mails zurück:

```
C: A282 SEARCH SINCE 1-NOV-2001 FROM "ADAM"
S: * SEARCH 2 84 882
S: A282 OK SEARCH completed
```

Beendet der Client mit dem Kommando LOGOUT die Verbindung, wechselt der Server in den "Update State" und führt noch anstehende Änderungen aus.

Daneben gibt es eine Reihe weiterer Befehle im "Authenticated State" und "Selected State". Eine Beschreibung aller Funktionen und Befehle würde den Rahmen dieses Artikels sprengen. An dieser Stelle können wir nur auf die rund 80 Seiten starke RFC2060 verweisen.

2.3.4 Beispiel für das Abrufen einer Mail

In diesem Beispiel sehen Sie den Ablauf einer IMAP4-Verbindung. Der Client identifiziert sich gegenüber dem Server, wählt eine Mailbox aus und lädt den Header einer Nachricht herunter.

```
S: * OK IMAP4 Service Ready
C: a001 login eva ahd635d
S: a001 OK LOGIN completed
C: a002 select inbox
S: * 18 EXISTS
S: * FLAGS (\\Answered \\Flagged \\Deleted \\Seen \\Draft)
S: * 2 RECENT
S: * OK [UNSEEN 17] Message 17 is first new message
S: * OK [UIDVALIDITY 3857529045] is first new message
S: a002 OK [READ-WRITE] SELECT completed
C: a003 fetch 12 rfc822.header
S: * 12 FECH (RFC822.HEADER {346}
S: Date: Wed, 10 Dec 2001 02:23:25 -0700 (PDT)
S: From: Adam <adam@xyz.de>
S: Subject: Beispiel für eine IMAP4-Verbindung
S: To: Eva <eva@test.de>
S: Message-Id: <9307191947AA00502.Adam@xyz.de>
S: Mime-Version: 1.0
S: Content-Type: TEXT/PLAIN; CHARSET=iso-8859-1
S: )
S: a003 OK FETCH completed
C: a004 LOGOUT
S: * BYE IMAP4 server terminating connection
S: a004 OK LOGOUT completed
```

Nachdem der Mail-Client über TCP eine Verbindung zum SMTPServer aufgebaut hat, wartet er auf einen Begrüssungstext des Servers. Im nächsten Schritt identifiziert sich der Client mit dem Kommando LOGIN, als Argument übergibt er den Benutzernamen und das Passwort. Nach dem Auswählen der Mailbox sendet der Server einige Informationen wie beispielsweise die Anzahl der ungelesenen Nachrichten. Mit dem Kommando FETCH fordert der Client den Header der Nachricht 12 an. LOGOUT beendet die Verbindung.

3 Anti-Spam für MTAs

In Massen verschickte Werbemails, auch Spam genannt, haben in den letzten Jahren immer mehr zugenommen. Einige Internet-Provider und E-Mail-Dienste lehnen Mails, die von bekannten Spammern stammen, bereits gänzlich ab. Derartige Massnahmen betreffen allerdings im schlimmsten Fall nicht nur den Mail-Account des Spammers, sondern sämtliche Post, die der betreffende E-Mail-Server versendet. Dabei ist es gar nicht so leicht, zwischen Werbemails und beispielsweise einem angeforderten Newsletter zu unterscheiden. So kann es passieren, dass die eine oder andere Mailing-Liste im Filter landet und ihre Empfänger nicht mehr erreicht.

Bei den Spammern ist es gängige Praxis, zum Versenden der Mails fremde SMTP-Server zu verwenden, so genanntes Relaying. Dies ist nicht nur in vielen Ländern illegal, sondern bringt für den Betreiber des betroffenen Servers unter Umständen zahlreiche Probleme mit sich: Dieser hat letztendlich das unnötige Datenvolumen zu bezahlen, und das plötzlich auftretende E-Mail-Aufkommen kann seine Infrastruktur lahm legen. Ausserdem ist es bei häufigem unautorisiertem Relaying möglich, dass der Server auf eine so genannte Blacklist gesetzt wird. Andere MTAs akzeptieren von diesem Server auf Grund der vielen Spammings keine Mails mehr. Dies bedeutet im schlimmsten Fall, dass sich von diesem SMTP-Server keine Nachrichten mehr versenden lassen.

Um den zweifelhaften Marketingmassnahmen der Versender erfolgreich Einhalt zu gebieten, ist es erforderlich, das Versenden derartiger Massenmails schon im Vorfeld zu verhindern.

3.1 Verhindern von Relaying

Ein SMTP-Server sollte in der Lage sein, unautorisiertes Relaying zu erkennen und abzulehnen. Beim Versenden einer Mail gibt es dazu vier Elemente zur Identifizierung von Sender und Empfänger mit unterschiedlichem Sicherheitsgrad:

Identifizierung von Sender und Empfänger einer Mail

Identifizierung	Beschreibung
HELO Hostname	Es kann keiner oder jeder beliebige Hostname angegeben werden.
MAIL From:	Der Client kann jede beliebige Adresse angeben.
RCPT To:	Dies muss eine korrekte Adresse sein.
SMTP_CALLER	IP-Adresse des Client.

Die ersten beiden Punkte (HELO und MAIL) können beliebige Angaben enthalten. Auf diese Angaben sollte man sich somit nicht verlassen. Daher sollte der Server das Relaying anhand des folgenden Algorithmus erlauben:

- Die Empfänger-Adresse der Mail gehört zu den "eigenen" Domains.
- Mails an die Empfänger-Domain werden grundsätzlich angenommen und weitergeleitet (MX Record).

- Die IP-Adresse des Client ist bekannt und autorisiert.

Damit lassen sich zumindest die meisten unautorisierten Relaying-Versuche verhindern.

3.2 Weitere Möglichkeiten

Eine weitere Möglichkeit gegen unautorisiertes Mail-Relaying ist die SMTP-Authentifizierung. Der Mailserver identifiziert den Client anhand von Zugangsdaten und erlaubt nur mit diesen eine Weiterleitung. Dies passiert bei jedem Versand. Das Verfahren entspricht dem Quasi-Standard RFC2554, der von gängigen Mail-Clients wie Microsoft Outlook und Netscape Messenger unterstützt wird.

3.3 SMTP after POP

Viele Provider setzen die SMTP-Authentifizierung nicht ein, da dies nicht von jedem System unterstützt wird. Stattdessen wird das Mail-Relaying dynamisch freigeschaltet. Dabei werden Nachrichten wie bisher per POP3 oder IMAP4 vom Server abgerufen. Hierfür identifiziert sich der Client gegenüber dem Server mit einem Benutzernamen und Passwort und überträgt auch seine IP-Adresse. Das System erlaubt nun dieser IP-Adresse den Versand von E-Mails für eine bestimmte Zeit. Bei "SMTP after POP" muss also zumindest einmal vor dem Senden einer Mail das Postfach abgefragt worden sein.

Weitere Informationen zu unerwünschten Werbemails finden Sie im Artikel "Kampf dem Spam".

4 TLS: Sicherheit beim Mailen

Ein Nachteil von SMTP ist die Sicherheit. Die Mail Transfer Agents kommunizieren untereinander im "Klartext". In den meisten Fällen geht die Übermittlung über einen oder mehrere Router. Dieser kann im schlimmsten Fall den gesamten Datenverkehr zwischen Client und Server mitprotokollieren und auswerten.

Um einen sicheren Mailversand zur gewährleisten, gibt es die Möglichkeit, eine SMTP-Verbindung per TLS-Verschlüsselung aufzubauen. **TLS (Transport Layer Security)** ist eine Weiterentwicklung des bekannteren SSL (Secure Sockets Layer). Details zu "SMTP over TLS" finden Sie in RFC2487.

Über das ESMTP-Kennwort STARTTLS teilt der Server dem Client beim Verbindungsaufbau mit, dass die Verschlüsselung unterstützt wird. Falls der Client Verschlüsselung nutzen möchte, sendet er das Kommando STARTTLS ohne Parameter. Beide Rechner starten daraufhin die Verschlüsselung. Hier sehen Sie ein Beispiel für eine SMTP-Verbindung per TLS:

```
S: 220 test.de SMTP server ready
C: EHLO xyz.de
S: 250-xyz.de, pleased to meet you
S: 250 STARTTLS
S: 220 Go ahead
C: <Start der Verschlüsselung>
S: + C: <Verschlüsselung wird abgesprochen>
C: <Client sendet Kommandos zur Bearbeitung von Mails>
...
```

Auch bei POP und IMAP besteht das Problem, dass insbesondere die Authentifizierungsdaten offen über das Internet gesendet werden. Daher wurde TLS auch für POP und IMAP eingeführt. Details hierzu finden Sie in RFC2595. (kpf)

5 Glossar

MUA (Mail User Agent)
Das eigentliche E-Mail-Programm

MTA (MAIL TRANSFER AGENT)
Dieser Dienst ist für die Weiterleitung und Zustellung von E-Mails zuständig. Nach Erhalt einer Mail von einem Mail User Agent (dem eigentlichen E-Mail-Programm) oder einem anderen MTA analysiert der MTA die Mail und liefert sie entweder an den lokalen User (bzw. seinen Mail User Agent) aus oder leitet sie an einen anderen MTA weiter, dabei wird meist das SMTP-Protokoll verwendet. Der im Internet immer noch am häufigsten benutze MTA ist "Sendmail".

MDA (MAIL DELIVERY AGENT)
Dieser Dienst legt die E-Mails in die verschiedenen E-Mail-Postfächer ab. In einigen Fällen wird für eine bessere Kommunikation zwischen dem MTA und dem MDA das Local Mail Transfer Protocol (LMTP) verwendet.

MRA (MAIL RETRIEVAL AGENT)
Dieser Dienst holt die gespeicherten E-Mails vom Mailserver (genauer dem MDA) ab und speichert sie auf dem lokalen Rechner. Meist werden die Protokolle POP3 oder IMAP unterstützt.

MAILFILTER
Auf einigen Mailservern werden Blacklists, Spamfilter, Anti-Virus-Programme und andere Filter eingesetzt.

MAPI
Abkürzung für "Messaging Application Programming Interface". Von Microsoft definierte Schnittstelle, mit der von jeder Windows-Software aus E-Mails verschickt werden können. Das Dokument, an dem gerade gearbeitet wird, wird dabei als Attachment angehängt. Zusätzlich sind in MAPI die Standard-Benutzerschnittstellen und Standard-Bedienerführungen definiert.