

Anwendungen der Internettechnologie

aus dem Seminar Internettechnologie

Prof. Dr. H. Göttler

T. Gottron

von Franz Susemichel

Inhaltsverzeichnis:

1	Remote Zugriff.....	3
1.1	Telnet.....	3
1.2	Secure Shell (SSH).....	5
1.3	Virtual Private Network (VPN).....	5
2	File Transfer Protocol (FTP).....	6
2.1	Das FTP – Modell:	6
2.2	Aufbau einer FTP – Verbindung:.....	7
2.3	Konfiguration der Datenübertragung:	7
2.4	FTP – Kommandos:	8
3	E-Mail.....	9
3.1	Geschichte und Entstehung der E-Mail.....	9
3.2	Verwendete Technologie.....	10
3.2.1	SMTP	10
3.2.2	POP.....	14
3.2.3	IMAP	14
3.3	Aufbau einer E-Mail.....	15
3.3.1	Ähnliche Konzepte.....	15
4	Newsgroup	16
4.1	NNTP	16
4.1.1	NNTP-Befehle.....	16
4.2	Usenet.....	17
5	Quellenangabe.....	18

1 Remote Zugriff

Zu Beginn des ARPANet entstand die Problematik, dass die zur Rechnerkonfiguration notwendigen Techniker sowohl an der West- als auch an der Ostküste der USA benötigt wurden. Als Reaktion darauf wurde ein Netzwerkprotokoll entwickelt, welches die Konfiguration eines Rechners von der ganzen Welt aus ermöglichte.

1.1 Telnet

Mit Telnet wird sowohl das Telnet-Protokoll als auch das dazugehörige Programm beschrieben. Das Telnet-Protokoll simuliert ein Virtuelles Terminal auf dem Host-Rechner. Somit wird der gesamte Funktionsumfang des Host auf das lokale Terminal abgebildet. Dabei ist zu beachten, dass sämtliche Befehle zwar standardisiert wurden, allerdings nicht zwingend umgesetzt werden müssen. So müssen z.B. bei einer Remote -Verbindung nur die Funktionen bereitgestellt werden, die auch lokal verfügbar sind. Zur Bereitstellung der Telnet-Verbindung wird serverseitig immer Port 23 verwendet. Die wichtigsten Befehle des Telnet-Protokoll sind:

- Interrupt Process (IP): Mit diesem Befehl ist es möglich, laufende Prozesse zu unterbrechen. Dies ist z.B. notwendig, wenn der Prozess in einer Endlosschleife ist. Der Befehl ist besonders wichtig, da er häufig von Programmen genutzt wird, welche mittels Telnet kommunizieren.
- Abort Output (AO): Durch das Senden von AO wird verhindert, dass die Ausgabe auf dem lokalen Terminal angezeigt wird. Lediglich ein "prompt" signalisiert die Bereitschaft des Terminal.
- Are You There (AYT): Mit diesem Befehl kann der Benutzer überprüfen, ob das System abgestürzt ist.
- Erase Character (EC): Hiermit lassen sich Fehler bei der Eingabe löschen. Dabei wird der "print buffer" geleert.
- Erase Line (EL): EL löscht eine komplette Zeile der Eingabe. Es wird ebenso wie EC zur Korrektur verwendet.

Da es bei einer Remote -Verbindung passieren kann, dass die Befehle für den Host gepuffert werden, wurde das SYNCH –Signal eingeführt. Durch das Senden des SYNCH – Befehl wird dem nachfolgenden Telenet-Kommando Priorität eingeräumt, und direkt auf dem Server ausgeführt. Um später an dieser Stelle fortfahren zu können, setzt der SYNCH-Befehl ein sog. DATA MARK (DM). Nachdem der SYNCH – Befehl durch ein STOP beendet wird, kann die ursprüngliche Verarbeitung ab dem DM fortgesetzt.

Die Telnet – Befehle werden durch ein vorangestelltes Zeichen (Interpret as Command (IAC)), sowie durch eine eindeutige Nummer identifiziert. Eine vollständige Liste findet man in dem RFC 854.

Ein weiteres Merkmal von Telnet ist die Erweiterbarkeit durch Optionen. Damit sich Sender und Empfänger auf eine Option verständigen können, wurden vier Befehle eingeführt:

- WILL: Dieser Befehl zeigt an, dass eine Option gewünscht ist, bzw. ist die Bestätigung, dass nun eine Option verwendet wird.
- WONT: Lehnt das Ausführen einer Option ab, bzw. zeigt an das die Option weiterhin verwendet wird.
- DO: Dieser Befehl fordert die andere Seite auf eine Option zu verwenden, bzw. zeigt an, dass man von der anderen Seite diese Option erwartet.
- DONT: Fordert die andere Seite auf eine Option zu beenden, bzw. zeigt an, dass eine Option nicht länger verwendet wird.

Die möglichen Optionen sind:

- Transmit Binary: Diese Option ist wichtig, da einige Programme, welche auf Telnet basieren, nur so in der Lage sind die Daten richtig zu deuten. Durch ein IAC lassen sich allerdings weiterhin Telnet-Befehle einflechten.
- Echo: Da heutzutage Telnet nicht mehr nativ genutzt wird, ist standardmäßig keine Ausgabe der Telnet-Befehle auf einem Terminal vorgesehen. Für den Fall, das die

verwendeten Telnet- Befehle kontrolliert werden müssen, erzeugt ECHO diese Ausgabe.

- **Suppress-Go-Ahead:** Da die heutigen Rechner, nicht wie die ursprünglichen Terminals, über eine vollduplex Verbindung verfügen, gibt es die mögliche diese zu simulieren¹.
- **Status:** Mit diesem Befehl lassen sich Statusmeldungen über die aktuell verwendeten Optionen zurückgeben.
- **Timing-Mark:** Mit Hilfe dieser Option ist es möglich falsche Eingaben abzufangen. Kann der Server eine Eingabe nicht interpretieren, sendet er ein DO TIMING-MARK und eine Fehlermeldung. Jede weitere Eingabe wird verworfen, bis er ein WILL TIMING-MARK des Client erhält.

Die Standardliste für Optionen enthält lediglich 255 Einträge. Sollten jedoch mehr Optionen benötigt werden, besteht die Möglichkeit ein "EXTENDED-OPTION-LIST" zu senden, und somit weitere 255 Einträge zu erhalten. Dabei wird der Code der neuen Option übergeben.

1.2 Secure Shell (SSH)

Da Telnet sämtliche Kommandos (auch Benutzernamen und Passwörter) in Klartext sendet, wurde ein Protokoll entwickelt, das den Versand von verschlüsselten Nachrichten ermöglichte.

Die letzte Version SSH 2 sendet über eine sichere, mit 128-Bit verschlüsselte, Verbindung und hat Telnet fast vollständig abgelöst. Anwendung findet das SSH-Protokoll z.B. in SCP, SFTP oder aber auch dem SSH-Client PuTTY.

1.3 Virtual Private Network (VPN)

Eine andere Möglichkeit des Remotezugriff bietet VPN. Über VPN wird eine Verbindung zwischen Client und Server getunnelt. Dieser Tunnel wird üblicherweise verschlüsselt. Als verbindendes Netzwerk wird meist das Internet verwendet. VPN wird häufig verwendet, um Außendienstmitarbeitern Zugriff auf das Firmennetzwerk zu ermöglichen.

¹ Leider habe ich hierzu keinerlei Verwendungsmöglichkeiten in der Praxis gefunden

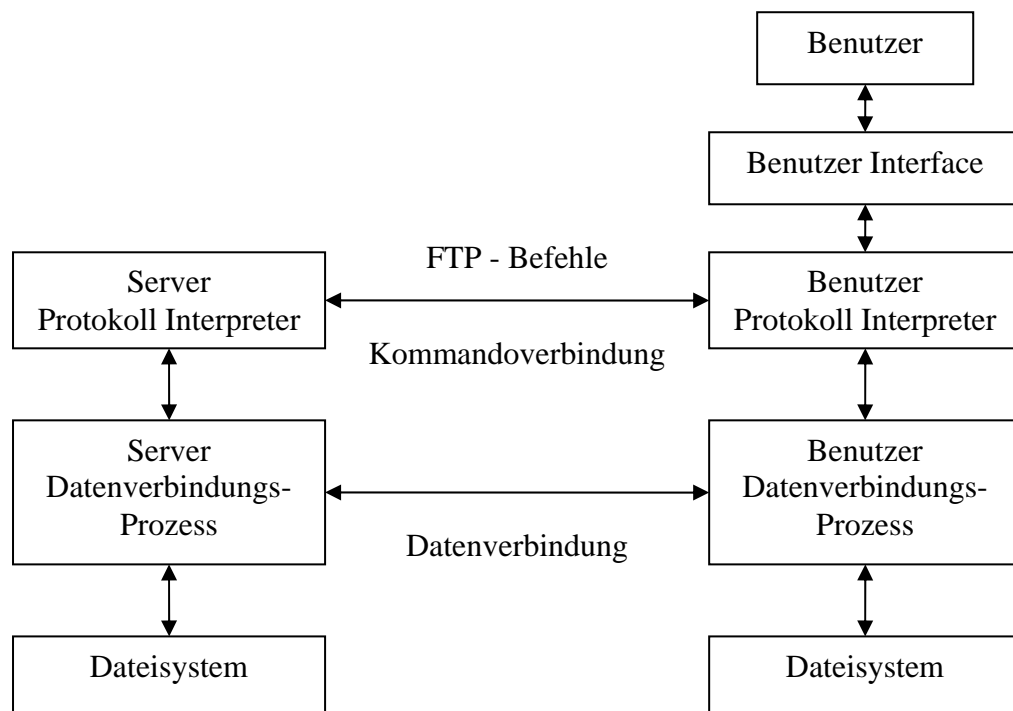
2 File Transfer Protocol (FTP)

Da zu Beginn des ARPANet keinerlei standardisierte Dateisysteme existierten, wurde FTP mit folgenden Zielen entwickelt:

- Gleichzeitige Nutzung von Daten durch verschiedene Quellen
- Vereinfachung des Zugriffs auf Remote-Systeme
- Bereitstellung einer einheitlichen Schnittstelle auf Dateisysteme
- Bereitstellung eines effizienten und ausfallsicheren Dateizugriff

Im Gegensatz zu Telnet basiert das FTP auf zwei Verbindungen. Über Port 21 werden die Kommandobefehle übertragen und über Port 20 die Daten. Beide Verbindungen basieren genau wie Telnet auf einer 8-Bit breiten Verbindung.

2.1 Das FTP – Modell:



2.2 Aufbau einer FTP – Verbindung:

1. Im ersten Schritt startet der Benutzer-PI den Aufbau der Kommando-Verbindung via Telnet.
2. Nachdem die Kommando-Verbindung aufgebaut wurde, erzeugt der Benutzer-PI FTP – Befehle, welche an den Server-DP übertragen werden. Diese Befehle enthalten alle Angaben um eine Daten-Verbindung zum Benutzer-DP herzustellen. Die Antwort auf die Befehle sendet der Server-PI.
3. Um die Daten-Verbindung herstellen zu können, muss der Benutzer-DP sich im "listen" – Modus befinden. Standardmäßig verwendet wird beim Benutzer für die Daten-Verbindung der gleiche Port wie für die Kommando-Verbindung verwendet. Es besteht von seitens des Benutzers aber die Möglichkeit diese zu ändern. Die Verwaltung der Daten-Verbindung ist allerdings Aufgabe des Servers.

2.3 Konfiguration der Datenübertragung:

- Um die Datenübertragung zwischen Servern mit unterschiedlichen Dateisystem zu ermöglichen, verfügt FTP über verschiedene Einstellungsmöglichkeiten:
 - Repräsentationstyp: Durch diesen Typ wird festgelegt, welches Format zur Übertragung der Dateien verwendet wird. Standardmäßig wird das NVT-ASCII-Format verwendet.
 - Dateistruktur: Der Benutzer hat die Möglichkeit, die Dateistruktur mit anzugeben. Damit soll die Übertragung zwischen verschiedenen Dateisystemen verbessert werden. Wird die Dateistruktur nicht implizit angegeben, werden die Daten als kontinuierlicher Datenstrom eingelesen.
- Nicht nur die Dateistruktur lässt sich anpassen, auch ein für die jeweilige Situation optimaler Übertragungsmodus lässt sich einstellen:
 - Stream Mode: Im Stream-Modus gibt es keinerlei Einschränkung bzgl. des verwendeten Repräsentationstyp oder der

verwendeten Dateistruktur. Die Bedingung zur Beendigung der Verbindung, hängt von der Dateistruktur ab.

- **Block Mode:** In diesem Modus werden die Dateien in Blöcken übertragen. Jedem Datenblock geht ein Header voraus, in welchem die Datenlänge sowie ein Code gespeichert sind. Dieser Code kann zur Fehlerkontrolle oder zur Beendigung der Verbindung verwendet werden.
- **Compressed Mode:** Bei diesem Übertragungsmodus werden gleiche Datenblöcke nur jeweils einmal übertragen. Vor der Übertragung eines solchen Blocks wird ein Header gesendet, der die Anzahl der Wiederholungen angibt.

Um eine Verbindung nach einem ungewollten Abbruch wieder aufnehmen zu können, müssen die Modi "Block" oder "Compressed" verwendet werden. Beim Block Mode kann im Header eine Markierung gesetzt werden, beim Compressed Mode ist es möglich die Stelle über die Dateistruktur wieder zu finden.

2.4 FTP – Kommandos:

Um die Dateiverwaltung komfortabler zu gestalten stellt FTP einige Befehle bereit:

- **Zugangsparameter:** Mit diesen Befehlen lassen sich Zugangsinformationen wie Benutzername, Passwort oder der Accountname angeben. Außerdem lassen sich Parameter wie das gewählte Verzeichnis etc. einstellen.
- **Übertragungsparameter:** Mit diesen Befehlen lassen sich alle, für die Übertragung relevanten Einstellungen tätigen (z.B. Übertragungsmodus).
- **Zugriffsparameter:** Diese Befehle regeln den Zugriff auf das Dateisystem, und werden (wenn auch mit unterschiedlichen Bezeichnungen) auch von vielen Betriebssystemen verwendet(z.B. Make directory, delete).

Bei FTP gibt es, genau bei SMTP, verschiedene Antworten, um einen empfangen Befehl zu bestätigen.

3 E-Mail

3.1 Geschichte und Entstehung der E-Mail

Die Entstehung der E-Mail geschah parallel zur Entwicklung des ARPANet. Da die damaligen Kommunikationsmöglichkeiten entweder zu langsam (Post), zu teuer (Telex) oder zu unsicher (Telegramm) waren, wurde eine neue Möglichkeit zur Kommunikation über größere Entfernungen gesucht. Die Idee, auf welcher die E-Mail basierte, war eigentlich nicht neu. Die



Geschichte der E-Mail beginnt Ende 1971. Der Erfinder der E-Mail, wurde dabei von der Möglichkeit, Texte zwischen zwei Mainframesystemen zu verschicken inspiriert. Um die Kommunikation zwischen den Rechnern des ARPANet zu ermöglichen schrieb Tomlinson zwei Programme, welche die Übermittlung von Nachrichten, sowie deren Einsortierung in ein Postfach ermöglichten. Beide Programme basierten auf einer frühen Version des File Transfer Protocol (FTP). Nachdem die Entwicklung innerhalb weniger Wochen abgeschlossen war, sendete Tomlinson Anfang 1972 die erste E-Mail.

Den Erfolg den Tomlinson mit der Erfindung der E-Mail hatte war überwältigend. Innerhalb kürzester Zeit verbreitete sich die E-Mail unter den Nutzern des ARPANet (vorwiegend Wissenschaftler und Militär). Bereits nach 2 Jahren wurden über 75 Prozent des Datenaufkommen im ARPANet durch E-Mail-Versand verursacht.

Das erste Programm zu Verwaltung von E-Mails wurde nur 4 Monate nach der ersten E-Mail geschrieben. Das erste Programm von Larry Roberts ermöglichte eine Auflistung der E-Mails sowie direkt auf eine E-Mail zu antworten. Der erste Alleskönner unter den E-Mails war eine Weiterentwicklung des Programms von Roberts, und enthielt eine komfortable Oberfläche, sowie der Möglichkeit Dateien anhängen zu können.

Den letzten Schritt zur eigenständigen Kommunikationsform schaffte die E-Mail 1977 als das FTP-Protokoll durch das Simple Mail Transfer Protokoll (SMTP) ersetzt wurde. Dies war notwendig, da E-Mails, welche über FTP versendet wurden, keinen einheitlichen Header hatten. Daher kam es zu Problemen bei der Adressierung und bei der Darstellung.

Der Erfolg der E-Mail lag in der extremen Verbesserung der Kommunikationsmöglichkeiten. Im Gegensatz zum damals sehr weit verbreiteten Fax, war die E-Mail sowohl schneller, günstiger als auch sicherer. Für die Wissenschaft wurde der Meinungs- und Wissensaustausch vereinfacht, dem Militär wurde eine ausfall- und abhörsichere Kommunikationsform geboten und die Wirtschaft bekam die Möglichkeit, sehr schnell und günstig einen großen Konsumentenkreis anzusprechen.

3.2 Verwendete Technologie

3.2.1 SMTP

Das Simple Mail Transfer ermöglicht den Versand einer E-Mail über das Internet. Bereits bei der Entwicklung von SMTP wurde darauf geachtet, dass das Protokoll eine Kommunikation über mehrere Netzwerke hinweg ermöglicht.

3.2.1.1 Die wichtigsten SMTP – Kommandos

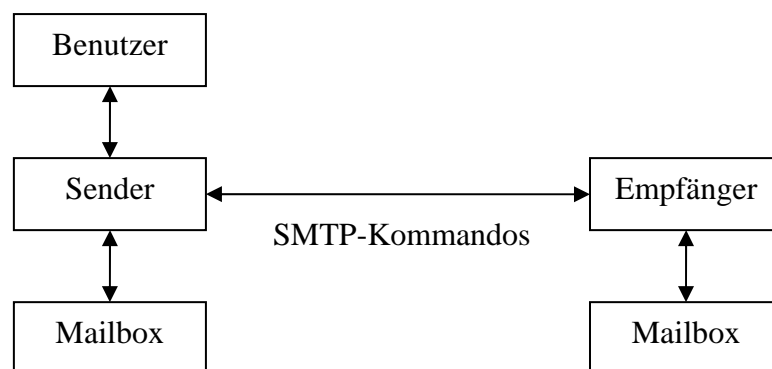
- **HELO:** Mit diesem Befehl identifiziert sich der Sender beim Empfänger. Dabei wird der Name des Senders übergeben. Durch ein "OK "-Signal akzeptiert der Empfänger und identifiziert sich damit beim Sender. Verläuft das HELO erfolgreich befinden sich sowohl Sender als auch Empfänger in ihrem Grundzustand.
- **MAIL:** Mit dem "MAIL"- Befehl wird der Beginn eines E-Mail-Versand gekennzeichnet. Als Parameter wird ein sog. "reverse-Path" übergeben, welcher die Liste der bereits passierten Hosts, sowie der Sender-Mailbox enthält. Im Falle eines Fehlers wird der "reverse-Path" dazu verwendet an den Sender eine Fehlermeldung zu schicken.
- **RCPT:** Mit diesem Befehl wird der Empfänger einer E-Mail gekennzeichnet. Soll die E-Mail an mehrere Empfänger verschickt werden, muss der Befehl mehrfach aufgerufen werden. Analog zum "MAIL" – Befehl, wird beim "RCPT" ein "forward-Path" angegeben. Dieser enthält die Hosts, die zum Erreichen der Empfänger-Mailbox notwendig sind.
- **DATA:** Mit diesem Befehl erhält der Empfänger das Signal, dass nun ein Datenstrom übertragen wird. Dieser enthält die eigentliche E-Mail.

Das Ende des Datenstroms wird durch eine, mit dem Befehl übertragene, Zeichenkette markiert. Verläuft das Kommando erfolgreich, fügt der Host einen Zeitstempel hinzu. Auf den Aufbau des Datenstroms wird später näher eingegangen.

- **SEND:** Der "SEND" – Befehl funktioniert ähnlich wie "MAIL", mit dem Unterschied, dass für "SEND" eine aktive Verbindung (mit direkter Ausgabe) notwendig ist. Ebenfalls ähnlich sind "SOML" (Send-or-Mail) und "SAML" (Send-and-Mail). Bei "SOML" wird erst "SEND" verwendet. Ist dies nicht erfolgreich wird die Nachricht mit "MAIL" übermittelt. "SAML" dient zur Verteilung an mehrere Empfänger, wobei bei den Aktiven "SEND" und bei den inaktiven "MAIL" verwendet wird.

Zusätzlich zu diesen Befehlen, gibt es noch einige Ergänzungen, welche z.B. das Auswerten einer Mailing-Liste ("EXPN") oder der Verifizierung eines Empfängers ("VRFY") dienen.

3.2.1.2 Das SMTP - Modell



SMTP ermöglicht das Versenden von E-Mails zwischen zwei oder beliebig vielen Servern. Dabei nimmt jeweils ein Server die Rolle des Senders oder des Empfängers ein. Die Kommunikation zwischen den Servern wird durch das SMTP definiert.

3.2.1.3 Der Ablauf des SMTP

Die Grundfunktion des SMTP beinhaltet das Versenden von E-Mails. Um eine E-Mail zu versenden wird zuerst das MAIL – Kommando ausgeführt. Der Empfänger wird so darüber informiert, dass eine E-Mail versendet werden soll. Der Empfänger muss den Erhalt des Kommandos bestätigen, indem es ein "OK 250" sendet.

Im nächsten Schritt sendet der Sender den RCPT - Befehl. Damit übermittelt der Sender den Empfänger der E-Mail. Dieser wird im "forward-Path" angegeben. Sind mehrer Empfänger vorgesehen, muss das RCPT mehrmals gesendet werden. Ist der Empfänger der E-Mail dem Empfänger-Server bekannt, sendet dieser ein "OK 250", ansonsten ein "550 failure". Allerdings hat der Empfänger-Server zusätzlich die Möglichkeit die Nachricht selbst weiterzuleiten ("251 user not local") oder dem Sender-Server einen anderen Server zu empfehlen ("551 user not local"). Dafür ist aber in beiden Fällen die Kenntnis einer alternativen Route vorausgesetzt. Akzeptiert der Empfänger das RCPT löscht er seinen Eintrag in der "forward-Path" -Liste und fügt seine Kennung in die "source-Path" –Liste ein. Ist die E-Mail nicht zustellbar löscht der Empfänger die "source-Path" –Liste komplett und trägt diese in die "forward-Path" –Liste ein. Die E-Mail geht also zurück an den Absender.

Der nächste Schritt ist der DATA –Befehl. Als Parameter wird ein String übertragen, welche das Ende des Datenstroms markiert. Wenn der Empfänger-Server den DATA – Befehl akzeptiert, sendet er die Bestätigung "354". Danach werden die Daten übertragen. Das Ende wird wieder mit einem "OK 250" bestätigt.

Ist der Empfänger der E-Mail dem Sender nicht bekannt, kann er eine Anfrage an den Empfänger-Server stellen. Mit dem "Verify" – Befehl bekommt der Sender-Server, falls vorhanden, die Adresse des E-Mail Empfängers mitgeteilt. Mit dem "Expand" – Befehl kann der Sender nach den Mitgliedern einer Mailing-List fragen. Der Empfänger-Server gibt im Erfolgsfall die E-Mail-Adressen der Mitglieder zurück, und kann optional einen Benutzernamen angeben.

Die Identifizierung von Sender und Empfänger geschieht über den HELO –Befehl. Das Ende einer Verbindung wird durch ein QUIT gekennzeichnet.

Beispiel:

S: "HELO" informatik.uni-mainz.de

E: "250 OK" zdv.uni-mainz.de

S: "MAIL FROM": test@informatik.uni-mainz.de

E: "250 OK"

S: "RCPT TO": [<test1@zdv.uni-mainz.de>](mailto:test1@zdv.uni-mainz.de)

E: "250 OK"

S: "RCPT TO": [<test2@zdv.uni-mainz.de>](mailto:test2@zdv.uni-mainz.de)

E: "251 User not local; will forward to [<test10@zdv.uni-mainz.de>](mailto:test10@zdv.uni-mainz.de)"

S: "RCPT TO": [<test3@zdv.uni-mainz.de>](mailto:test3@zdv.uni-mainz.de)

E: "550 No such user here"

S: "DATA"

E: "354 Start mail input; end with <Zeichenfolge>"

S: "blablabla"

S: "blubb"

S: <Zeichenfolge>

R: "250 OK"

S: "QUIT"

In diesem Beispiel würden "test1" und "test10" (über "test2") die E-Mail erhalten. Da "test3" nicht bekannt ist, würde hier keine E-Mail versandt werden.

Ein großes Problem von SMTP war, dass es keinerlei Autorisierungsmöglichkeiten vorsah. Dieses Problem wurde durch Weiterentwicklungen wie SMTP-Auth oder SMTP-After-POP behoben. Um eine Manipulation von E-Mails zu verhindern wurden Verschlüsselungsmethoden wie z. B. "Pretty Good Privacy" (PGP) entwickelt.

3.2.2 POP

Das "Post-Office-Protocol" (POP) wurde notwendig, da SMTP nur bei ständiger Internetverbindung zufrieden stellend funktionierte. Durch die stetig wachsende Zahl an Benutzern ohne ständige Internetanbindung, war eine neue Lösung notwendig. Das POP bietet die Möglichkeit sich für kurze Zeit am Mailserver anzumelden, die E-Mails lokal zu speichern und auf dem Server zu löschen. Ein großes Problem des POP ist, dass alle E-Mails lokal gespeichert werden.

3.2.3 IMAP

Um die Problematik von POP zu beseitigen wurde das "Instant Message Access Protocol (IMAP) entwickelt. Im Gegensatz zu POP, werden bei IMAP alle E-Mails direkt auf dem Mailserver verwaltet. Somit ist der Zugriff auf die E-Mails jederzeit gewährleistet. Der erhöhte Leistungsumfang ist jedoch für viele Server problematisch, da IMAP zum Teil sehr lange TCP -Verbindungen herstellt.

Sowohl bei POP als auch bei IMAP wird zum Versenden der E-Mails weiterhin das SMTP verwendet.

3.3 Aufbau einer E-Mail

Eine E-Mail besteht immer aus einem Kopf ("Header") und einem Rumpf ("Body"). Im Kopf der E-Mail werden alle Informationen wie Empfänger, Absender, verwendete Route oder der Betreff gespeichert. Der Rumpf enthält die eigentliche Nachricht. Um Kopf und Rumpf eindeutig voneinander zu trennen, wird eine Leerzeile eingefügt.

Beispiel:

- Received: by Server Mo, 14.11. 06:00:00
- Date: So, 13.11. 18:00:00
- Subject: Seminar-Abgabe
- Message-ID: 12153462@students.uni-mainz.de
- From: susefran@student.uni-mainz.de
- To: gotti@informatik.uni-mainz.de
- CC: goettler@informatik.uni-mainz.de
- User-Agent: Outlook2003
- In-Reply-To: 123412341251@informatik.uni-mainz.de
- Leerzeile um den Header vom Body zu trennen
- Body

3.3.1 Ähnliche Konzepte

Der große Erfolg der E-Mail führte dazu, dass es mit steigender Bandbreitenkapazität und steigender Nutzerzahl, immer mehr Alternativen gab. So lässt sich beispielsweise ein Chat auch mit dem SMTP-Befehl "SEND" realisieren. Andere Konzepte waren ICQ, VoiceMail oder Webcams.

Im kommerziellen Bereich existiert lediglich X.400 als Alternative zur E-Mail. In Deutschland wird dieses Produkt unter dem Namen Telebox 400 vertrieben.

4 Newsgroup

Newsgroups sind Online-Diskussionsforen und werden häufig auch als Schwarze Bretter oder Bulletin-Boards beschrieben. Newsgroups werden entsprechend ihren Themen sortiert und verwaltet. Um in einer Newsgroup etwas schreiben zu können wird ein sog. Newsreader benötigt.

4.1 NNTP

Das Network News Transfer Protocol ermöglicht die Erstellung, Verteilung und das Aufrufen von Nachrichtenartikeln (News). Die Artikel werden zusammen mit einer Beschreibung zentral gespeichert. Alle Artikel werden von einem Newsserver in dieser Zentrale verwaltet. Der Newsserver stellt die Schnittstelle zwischen Benutzer und der Datenbank dar. Die Verbindung funktioniert wie bei SMTP mittels Befehlen und Antworten als ASCII-Code.

4.1.1 NNTP-Befehle

- **ARTICLE:** Mit diesem Befehl bekommt der Benutzer einen Artikel zurückgeliefert. Die Identifizierung des Artikel geschieht entweder über eine ID (wird von einem Server fortlaufend vergeben) oder über eine Nummer (identifiziert einen Artikel innerhalb eines Thema).
- **GROUP:** Dieser Befehl liefert die Artikelnummer des ersten und letzten Artikel der gewählten Newsgroup zurück.
- **HELP:** Liefert einen Überblick über alle verfügbaren Befehle zurück.
- **IHAVE:** Mit diesem Befehl signalisiert der Client, dass er einen neuen Artikel einstellen möchte. Der Server prüft anhand der übergebenen ID ob er die Nachricht akzeptiert. Verläuft dieser Test positiv, sendet er dem Client eine Antwort, in der er zum Senden der gesamten Nachricht auffordert.
- **LAST:** Liefert den letzten Eintrag einer Newsgroup zurück.
- **LIST:** Liefert eine Liste mit allen Newsgroups zum angegebenen Thema zurück. Es wird jeweils die Gruppe zusammen mit dem letzten und ersten Artikel zurückgeliefert.

- NEWSGROUP: Dieser Befehl liefert alle Newsgroups zurück, die bis zu einem bestimmten Datum angelegt wurden.
- NEWNEWS: Funktioniert ähnlich wie NEWSGROUP und liefert alle bis zu einem bestimmten Zeitpunkt erstellten Artikel zurück.
- NEXT: Liefert den nächsten Artikel der aktuellen Newsgroup zurück. Ist bereits der letzte Artikel einer Newsgroup gewählt, wird ein Fehler zurückgegeben.
- POST: Prüft ob es erlaubt ist neue Artikel zu senden. Liefert der Server OK zurück, wird der neue Artikel versandt.
- QUIT: Beendet die Verbindung zwischen Server und Client.
- SLAVE: Zeigt an, dass eine Server-Server-Verbindung besteht. Server-Server-Verbindungen werden gegenüber Server-Client-Verbindungen bevorzugt behandelt.

4.2 Usenet

Der größte Zusammenschluss von Newsgroups ist das Usenet. 1979 als "Schwarzes Brett" in der Universität von North Carolina entstanden, umfasst es heutzutage bereits mehr als 20.000 Newsgroups.

Die einzelnen Newsgroup sind thematisch sortiert, und ermöglichen es somit dem Benutzer schnell und einfach Beiträge zu finden und zu veröffentlichen.

In der Regel unterliegen Newsgroup keinerlei Einschränkungen. Es ist allerdings möglich einen Postmaster einzusetzen, der die Möglichkeit besitzt unerwünschte Einträge zu löschen.

5 Quellenangabe

- Requests for Common
- www.mediensprache.de
- <http://cartoon.iguw.tuwien.ac.at/fit/2001/welcome.html>
- <http://www.michaelkaul.de/Geschichte/geschichte.html>