Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Lehrstuhl: Rechnernetze und Telematik

WS 2007/08

Seminararbeit

Electronic Mail (SMTP, POP, IMAP, MIME) and Spam

Peter Rabiański

15. Februar 2008

Betreut von Prof. Dr. Christian Schindelhauer

Abstract

Es wird erläutert, wie eine E-Mail vom E-Mail-System abgearbeitet wird. Es wird außerdem erläutert mit welchen Techniken es zustande kommt und welche Server und welche Protokolle die einzelnen Aufgaben übernehmen. Dazu wird anhand eines Beispieles der Ablauf der verschickten E-Mail von Herstellung einer TCP Verbindung, der Prozess des Verschickens einer E-Mail, Clientkontrolle, Serverkontrolle bis zur Trennung einer TCP Verbindung gezeigt.

Da per Mausklick Tausende oder sogar Millionen von E-Mail -Nutzern erreicht werden, wird diese Art von Kommunikation besonders von unehrlichen Internetbenutzern ausgenutzt, die Werbung jeder Art an zufällige Leute verschicken. Es werden Techniken Vorgestellt, wie man sich inzwischen vor einer überproportionalen Anzahl von *Spam*(UBE) im Vergleich zu *Ham*(legitimer E-Mail) schützen kann.

Inhaltsverzeichnis

1 Introduction

E-Mail hat unser Leben stark beeinflusst und ist zum wichtigen Bestandteil des Computernutzens geworden. Innerhalb von wenigen Jahren ist die E-Mail eine der wichtigsten und vor allem eine der populärsten Applikationen geworden. Der Anteil von E-Mails gegenüber traditioneller Kommunikation (Post) ist am Anfang der XXI. Jahrhunderts schon deutlich größer geworden. Noch in der Mitte der neunziger Jahre die E-Mail einer nicht offiziellen Kommunikation diente, sieht es im Jahr 2008 deutlich anders aus. Dass die E-Mail nicht nur guten Zwecken dient, haben schon viele E-Maibenutzer erfahren.

2 Supports for Email

E-Mail erlaubt den Benutzer verschiedene kleine Notizen sowie große Nachrichten innerhalb von wenigen Sekunden an einzelne oder an *E-Mailisten* zu verschicken. Die E-Mail besitzt eine gute Eigenschaft, die bei anderen Internetdiensten nicht unterstützt wird. Sie erreicht sein Ziel auch dann, wenn das Ziel (Server) vorübergehend aus verschiedenen

Gründen nicht erreichbar ist. Die aufgehaltene Nachricht wird mittels *spooling* Technik bei nächster Gelegenheit verschickt, ohne dass der Absender es merkt. Dazu werden außer einer Kopie der Nachricht der Absender, Empfänger, Zielserver und die Zeit der Verschickung gespeichert. Im Hintergrund erstellt der *Client* eine TCP -Verbindung und verschickt eine E-Mail an den Zielserver. Die Aufgabe endet mit Erfolg, wenn der Zielserver eine Kopie der verschickten E-Mail im *spool* speichert. Dann löscht der *Client* die Kopie dieser E-Mail. Im Falle eines Misserfolges der Herstellung einer TCP- Verbindung wird nach 30 Minuten nochmal versucht die nicht verschickten E-Mails vom *spool* Speicher zu verschicken. Wenn eine E-Mail aus irgendeinem Grund nicht verschickt werden kann, kommt sie nach drei Tagen zum Absender zurück.

3 Mail Client

E-Mail-Client ist eine Software, die zusammen mit dem E-Mail-System interagiert. Die Software ist noch unter dem Begriff *Mail User Agent (MUA)* Die wichtigsten Elemente, die sie beinhaltet sind:

- E-Mail-Editor
- Adressbuch
- Speicher
- Kommunikationsmodul

Ein E-Mail-Editor erlaubt dem Benutzer eine E-Mail zu schreiben, sie nach Wunsch zu formatieren oder die Schreibfehler zu überprüfen. Adressbuch und Speicher gehören zu jedem E-Mail-Programm. Das Kommunikationsmodul ist die wichtigste Komponente des E-Mail-Programms. Sie ist dafür zuständig, die E-Mails zu verschicken und sie zu empfangen. Eine detaillierte Beschreibung ist im Abschnitt 5 und Abschnitt 6 zu sehen.

4 Email Architecture

Seit dem Einführen der E-Mail, sind einige E-Mail-Architekturen möglich.

- ein einfaches E-Mail-System
  Dieses System gehört zu den ältesten und benötigt keine Internetverbindung. Diese Art der Kommunikation wird dort benutzt, wo die Sicherheit die höchste Priorität

hat. Zu dieser kommunizierenden Computer wird noch ein E-Mail-Server verbiunden. Jede ausgehende Nachricht geht über den Server. Vom Server wird überprüft, ob der Empfänger überhaupt vorhanden ist. Die Nachricht wird erst dann von seinem *Client* aufgerufen, wenn er wieder erreichbar ist. Die Adressierung für diese Art der Kommunikation kann sehr vereinfacht werden. Je nach Größe des Netzwerkes und Anzahl der Anwender sind die Benutzernamen für die Adressierung ausreichend.

- einfache E-Mail-Systeme kommunizieren zusammen
  Es ist möglich, dass zwei Unternehmen ohne Internetverbindung kommunizieren. Zwei weit auseinander liegende Netzwerke werden mit zwei Modems und einer Telefonlinie verbunden. Die Adressierung muss etwas komplexer sein, als bei einer einfachen E-Mail-Architektur, weil es dazu kommen kann, dass die Anwender beider Netzwerke mit gleichen Benutzernamen versehen sind. Um Konflikte in der Kommunikation zu vermeiden, wird die Adressierung dieser Art verwendet:

  ```
  user-name@domain-name
  ```

  Die vorgeschlagene Adressierung erlaubt eine Erweiterung, weil jedes Netzwerk seinen eigenen Domänennamen besitzt.

- ein Benutzer kommuniziert über das Internet
  Das Prinzip der Kommunikation wurde nicht geändert, außer dass der Benutzter über das Internet kommuniziert. Auch in diesem Fall bleibt die Adressierung die aufwändigste Seite der Kommunikation. Jeder teilnehmende Partner der Kommunikation muss durch gültigen Benutzernamen und Domänennamen identifiziert werden. Diese Art der Kommunikation ist heutzutage am meisten verbreitet.

4.1 Email Address

Jede gültige E-Mail-Adresse besteht aus zwei Teilen, getrennt durch @ (*at*) Zeichen. Vorne befindet sich der Benutzername und hinten der Domänenname (das Ziel). Alle Benutzernamen müssen zu einem Domänennamen unterschiedlich sein.

4.2 Forwarding Software

Fast alle E-Mail-Server unterstützen das Weiterleiten (*Forwarding*). Es bedeutet, dass jede ankommende E-Mail, an eine oder an mehrere E-Mail-Adressen weiter verschickt werden kann. Der Server nimmt den Kontakt mit der Datenbank von *aliases* auf und gibt die E-Mail-Adresse des Empfängers an andere Adressen und schickt die E-Mails an

alle Teilnehmer aus dieser Liste. Es existiert *many-to-one* und *one-to-many alias*. Diese Eigenschaft gibt einem E-Mail-System eine zusätzliche Funktionalität. Ein Benutzer kann viele Benutzernamen haben oder ein Benutzername *alias* kann für eine Gruppe von E-Mail-Empfängern(*electronic mailing list*) dienen. Im ersten Fall bekommt eine Person eine E-Mail, im zweiten Fall bekommen alle Teilnehmer eine E-Mail, die in einer Mailingliste gespeichert sind. Ein E-Mail-Programm, welches *aliases* unterstützt sorgt dafür, dass jede ankommende, sowie ausgehende E-Mail über das *alias expansion and forwarding* führt.

5 TCP/IP Standart

5.1 RFC 2822

Der Standard gibt die Syntax für eine valide E-Mail. Der Standard wurde eingeführt um die Kommunikation zwischen heterogenen Systemen zu vereinfachen. Eine E-Mail wir als ein Text dargestellt, der zwei wichtige Bestandteile *hader* und *body* besitzt. Die Beiden sind durch eine leere Zeile getrennt. Der Syntax des *Header* ist durch den Standard genau festgelegt. In *Header* muss mindestens die Information *To:* eingegeben werden. Welche Zeichen erlaubt bzw. pflichtig sind, wird auch im RFC 2822 [4] Standard festgelegt. Der *Haeder* kann noch zusätzliche Zeilen wie *From:* oder *Replay-to:* enthalten. In diesem Fall kann der Absender einer E-Mail eine andere E-Mail-Adresse für den Absender eingeben. Auch wenn der Absender einer Nachricht eine Antwort erwartet, kann eine andere E-Mail-Adresse eingegeben werden, als die,die von der ursprünglichen E-Mail-Adresse verschickt wurde. Der *Body* kann dagegen vom Benutzer beliebig dargestellt werden.

6 Simple Mail Transfer Protocol (SMTP)

Ein Transfer Protokoll, der das Format für E-Mail-Übertragung festlegt, ist als *Simple Mail Transfer Protocol* bekannt. Das Format spezifiziert, wie eine Nachricht von *Client* den *Server* erreicht. Das Format schreibt allerdings nicht vor, wie eine E-Mail von der Benutzeroberfläche gezeigt wird und wie oft das System versuchen wird, eine E-Mail zu verschicken.

6.1 SMTP with Example

Das Beispiel zeigt den Ablauf von Befehlen während des Übertragens einer E-Mail.
Vom *Host* Namens *Alpha.edu* wird eine E-Mail an *Host* Namens *Beta.gov* verschickt.

Es wird ein Fall berücksichtigt, wo der Benutzer nicht bekannt ist und ein Fall, wo die verschickte Nachricht den Empfänger erreicht.

```
S: 220 Beta.gov Simple Mail Transfer Service Ready
C: HELO Alpha.edu
S: 250 Beta.gov

C: MAIL FROM <Smith@Alpha.edu>
S: 250 OK

C: RCPT TO:<Jones@Beta.gov>
S: 250 OK

C: RCPT TO:<Green@Beta.gov>
S: 550 No such user here
C: RCPT TO:<Brown@Beta.gov>
S: 250 OK

C: DATA
S: 354 Start mail input; end with <CR><LF>.<CR><LF>
C: ...sends body of mail message...
C: ...continues for as many lines as message contains
C: <CR><LF>.<CR><LF>
S: 250 OK

C: QUIT
S: 221 Beta.gov Service closing transmission channel
```

Figure 26.3 [1, Comer, chap.26]

6.2 Two-Stage Delivery Process

Die *two-stage delivery process* erlaubt das Übertragen von E-Mails, wenn es an fester Internetverbindung mangelt.

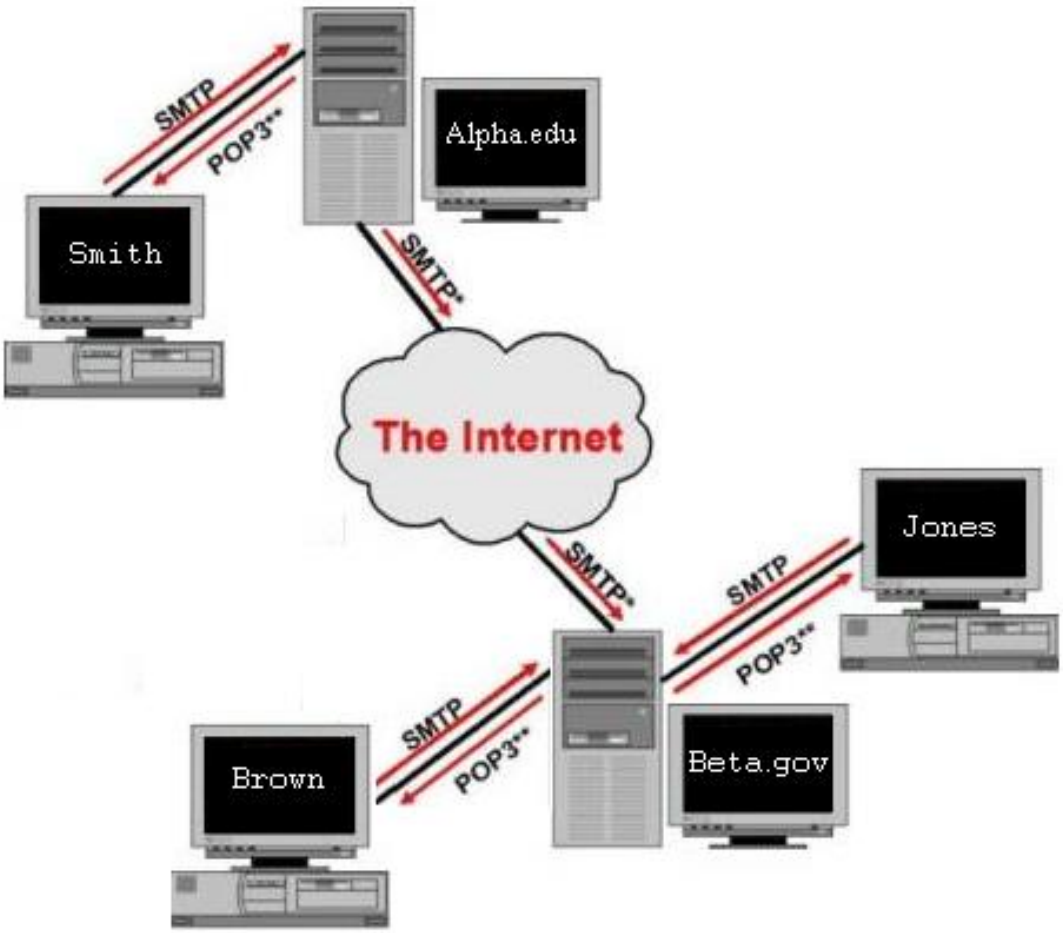

Abbildung 1: Email Comunication

- Es wird eine feste Internetverbindung gefordert. Der SMTP Server ist immer bereit die E-Mails zu empfangen.
- Der Benutzer verbindet den Computer mit dem Internet und kopiert die E-Mails aus dem festen Postfach auf den Computer.

6.3 SMTP Not Optimal

SMTP funktioniert nicht optimal, wenn ein Server auf dem Computer keine dauerhafte Internetverbindung hat. In diesem Fall wird eine andere Lösung gesucht. Alle in der Zeit ohne Internetverbindung verschickten E-Mails wären als nicht zugestellt betrachtet.
Für den Fall kommt mit der Hilfe *Post Office Protocol*.

6.4 MX Record (MX)

Die mit dem *SMTP* versandte E-Mail wird zum richtigen E-Mail-Server mit Hilfe von *DNS* geliefert. Es bedeutet, dass das Domain Name System (*DNS*) zu IP Adresse konvertiert wird. Es ist möglich einer Domäne mehrere E-Mail-Server zuzuweisen. Dieses Verfahren ist praktikabel, wenn ein oder mehrere E-Mail-Server z.B. wegen Wartungsarbeiten ausgeschaltet bleiben. Damit wird die Zustellung der E-Mail nicht gefährdet. Die *DNS* Server haben einen speziellen Eintrag *Mail eXchanger (MX)* der die E-Mail-Server nach Priorität ordnet und damit entscheidet welcher Server eine E-Mail-Nachricht für die Zustellung [6, TekGuard] übernimmt.

$$\underbrace{e-mail.com}_{DomainName}\ IN\ MX\ \underbrace{10}_{Priority}\ \underbrace{post.e-mail.com}_{MailServer}$$

7 Post Office Protocol v. Internet Massage Access Protocol

Zwei Protokolle mit gegenüberliegenden Wirkungen erlauben dem Benutzer auf die E-Mails zuzugreifen. Der *POP* sorgt dafür, dass die E-Mails heruntergeladen werden. Der *IMAP* erlaubt die Manipulation von E-Mails auf dem Server.

7.1 POP

Post Office Protocol (POP), die bekannteste Version *(POP3)* und die sichere Version *(POP3S)* erlauben das Übertragen von E-Mails von einem festen Postfach zu POP Cli-

ent auf dem Rechner. Auf die Forderung des POP-Servers wird eine TCP -Verbindung hergestellt. Bei der Verbindung zum POP- Server wird der Benutzer gefordert sich mit *login* und *password* zu authentifizieren. Nach einer erfolgreichen Authentifizierung werden die E-Mails vom festen Postfach auf die *POP Client* kopiert und zugleich aus dem festen Postfach gelöscht.
Auf dem Computer mit dem festen Postfach müssen zwei Server laufen. Der *SMTP Server*, empfängt die E-Mails und fügt sie in den festen Postfach hinzu. Es muss noch ein *POP* Server laufen, um die empfangenen E-Mails aus dem Postfach im Computer zu speichern. Es wird verlangt, dass jeder Server einen gültigen Zustand hinterlässt.

7.2 IMAP

Internet Message Access Protocol (IMAP), *IMAP4* oder eine sichere Version *IMAPS* erlaubt dem Benutzer das Lesen und die Manipulation von E-Mails. *IMAP Client* erlaubt die Manipulation von E-Mails aus verschieden Standorten und synchronisiert sie. *IMAP* erlaubt die Sicht des *Header* einer E-Mail ohne sie herunterzuladen. Das Protokoll gibt die Möglichkeit einen Teil der E-Mail herunterzuladen. Diese Eigenschaft ist bei langsamen Internetverbindungen von Vorteil.

8 Multipurpose Internet Mail Extensions (MIME)

Das Protokoll *Multipurpose Internet Mail Extension* wird für komplexere E-Mails benutzt. Eine komplexere E-Mail, die nicht aus *ASCII* Daten besteht, wird in *ASCII* kodiert und wie eine übliche E-Mail verschickt. Das *MIME* Protokoll unterstützt die *7-bit* Darstellung mit *base64* Kodierung *(Content-Transfer-Encoding:)*. Die Information über *MIME* nach 2822 Standard wird im Header einer E-Mail gespeichert. Daneben wird außer *From:*, *To:*, *MIME-Version:*, *Content-Type:* die Art der Datei und nach dem Schrägstrich *subtype* mit Endung einer verschickten Datei, *Content-Transfer-Encoding:* im Header aufgelistet. Um die verschickte Datei zu sehen, muss sie vom *base64* in *ASCII* dekodiert werden und mit dem geeigneten Programm geöffnet werden. Das *base64* wurde zur Kodierung-Standard ausgewählt, um die Kompatibilität einer dekodierten Nachricht zu bewerkstelligen. Der *MIME* Standard definiert sieben elementare *Content Types*. Zu denen zählen: *text, image, audio, video, application, multipart und message.*

8.1 MIME Multipart Messages

MIME Multipart Messages ist die Erweiterung vom *MIME* Protokoll. Die Teile einer E-Mail können aus vier möglichen Subtypen bestehen: *mixed, alternative, parallel, digest.* Eine E-Mail kann dadurch Textdateien, Musikdateien, Programmdatei oder alle diese Dateien zusammen enthalten.

- Subtyp *alternative* ist sinnvoll zu benutzen, wenn der Empfänger einer Nachricht ein anderes System benutzt
- Subtyp *mixed* wird empfohlen, wenn eine Nachricht viele Unterteile hat und jedes Teil muss anders dekodiert werden
- Subtyp *parallel* wenn zwei oder mehre Dateien zusammen abgespielt werden müssen, um die erwünschte Wirkung zu besitzen
- Subtyp *digest* - eine E-Mail kann andere E-Mails enthalten

```
From: bill@acollage.edu
To: john@example.com
MIME-Version: Multipart/Mixed; Boundary=StartOfNextPart

--StartOfNextPart
Content-Type: text/plain
Content-Transfer-Encoding: 7bit
John,
 Here is the photo of our research lab that I promised
 to send you. You can see the equipment you donated.

Thanks again,
Bill

--StartOfNextPart
Content-Type: image/gif
Content-Transfer-Encoding: base64
...data for the image...
```

Figure 26.6

9 Unsolicitated Spam Mail

Spam *Unsoliciated Bulk Email* ist ein großes Problem geworden, seitdem das Verschicken einer E-Mail eine einfache und günstige Aufgabenstellung geworden ist. Die allgemeine Zugänglichkeit des Internets für die Massen hat auch Nachteile gebracht. Jeder kann jedem schreiben. Dabei wird meistens nicht darauf geachtet, ob man das will oder nicht. Dass die E-Mail hauptsächlich für Werbezwecke geschickt wird, haben wir längst keinen Zweifell mehr. Inzwischen zeigen die Statistiken, dass etwa 94% der verschickten E-Mails Spam ist [5, SpamAssassin]. Es kann natürlich bei jedem unterschiedlich sein, abhängig davon, ob man seine E-Mail-Adresse angibt. Damit werden ca. 10 Milliarden Euro für die Verbindungskosten verloren. Man darf nicht vergessen, dass die Verbindungskosten nicht überall günstig sind. Es werden Satellitenverbindungen, Modemverbindungen oder UMTS, GPRS benutzt, wo keine herkömmliche Internetverbindung möglich ist. Sortieren von E-Mails während der Arbeit kostet natürlich Zeit und Ressourcen. Weil es immer mehr Spam gibt, wird man sich die Frage stellen müssen, ob die E-Mail Kommunikation sicher und vertraulich genug ist.

10 Spam Filtering

Es gibt inzwischen mehrere Methoden die Spam zu filtern. Man kann auf die kooperative Methode setzen, die mit dem guten Willen des Urhebers *Originator* verbunden ist. Der Urheber kooperiert mit dem Benutzer , in dem er eine signifikante Bezeichnung für seine E-Mails oder seine Identität des dem Adressat gibt. Der Urheber konnte seine E-Mails mit dem entsprechenden *Header* bezeichnen, damit der Benutzer es leichter hat, solche E-Mails zu sortieren. Man wird jedoch öfters mit Spam überschüttet, obwohl sich der Adressat von UBE abgemeldet hat. Diese Technik hat nur bei ehrlichen Urhebern eine Chance gut zu funktionieren. Eine weitere Idee wäre, vom Internetanbieter zu verlangen, die Missachter zu bestrafen. Vorher mussten entsprechende Richtlinien gemacht werden, um diese Art von Lösung zu bewerkstelligen. Es gibt inzwischen Versuche die Gesetze anzupassen, die bessere Kontrolle der Spammer zu ermöglichen zu können. Es ist allerdings nicht einfach politische Grenzen zu überschreiten. Nicht alle Länder haben daran Interesse mit dem Anti-Spam-Gesetz nachzukommen. Es wird mehr in heuristische Techniken gesetzt, die Spam effektiv und effizient zu filtern. Es gibt zwei wichtige Annäherungen zu dieser Technik:

11 Heuristic Filtering

11.1 Origin Filtering

Es werden vier Vorhänge von Herkunftfiltern *Origin Filtering* verwendet:

- Verweigerung von IP-Verbindung von den bekannten *UBE-Urheber* ;
  Die Technik ist bekannt als *black holing*. Diese Technik sorgt dafür, dass sie IP-Paketen von den Benutzer deren IP den IP Adresse der Absender entsprechen nicht versendet werden. Als Folge wird der Benutzter keine Nachricht vom gesperrten Absender bekommen können. Auch der Adressat wird nicht in der Lage sein mit dem Absender zu kommunizieren.
- Verweigerung der TCP-Verbindung von dem bekannten *UBE* Urheber in SMTP Server;
  Moderne SMTP Server haben die Möglichkeit nach IP Adressen oder Domänennamen des Absenders zu suchen. Wenn die IP Adresse mit einer aus der gesperrten übereinstimmt, kann der SMTP Server den Austausch weiterer SMTP Befehle verweigern. Die Filterung wird gleich nach dem Aufbau der TCP- Verbindung ausgeführt. Der Benutzer wird keine Nachricht vom Absender bekommen können. In vielen Fällen wird für den Adressat möglich, dem Absender (UBE-Urheber) die E-Mails zu schicken. Der Benutzer kann sich allerdings wundern, dass er nie eine Antwort auf seine E-Mails bekommt.
- Verweigerung der TCP-Verbindung von bekanntem UBE-Urheber nach *"MAIL FROM"* Befehl;
  Das Filtern wird nach Erhalten vom *"MAIL FROM"* Befehl ausgeführt. Es wird keine E-Mail verschickt. Es kann zu einer Situation kommen, dass der Benutzer eine E-Mail an den UBE-Urheber schickt, aber nie eine Antwort bekommt, ohne zu wissen warum es dazu gekommen ist.
- Verweigerung der TCP-Verbindung, wenn die IP Adresse mit dem Domänennamen nicht übereinstimmt;
  Der SMTP Server vergleicht, ob die IP Adresse mit dem Domänennamen übereinstimmt. Dann überprüft der Server, ob die IP Adresse mit der IP Adresse der TCP-Verbindung übereinstimmt. Wenn das nicht der Fall ist, kann der Server gleich nach dem *"MAIL FROM"* Befehl die Verbindung trennen. Die Folgen für den Benutzter bleiben wie beim Filtern nach dem *"MAIL FROM"* Befehl.

11.2 Classification

Allein die Klassifizierung in Spam und *Ham* (legitim E-Mail) ist nicht ausreichend, weil auch mit den besten Filtermethoden Fehler entstehen. Aus diesem Grund ist Bedarf entstanden vier Begriffe einzuführen, die genauer E-Mails klassifizieren. *True positive* (tp) wird für die richtig klassifizierte Ham benutzt. *True negative* (tn) wird für die richtig klassifizierte Spam benutzt. Es wird noch *false positive* (fp) für falsche Spam und *false negative* (fn) für falsch klassifizierte Ham. In manchen Publikation werden gegenüberliegende Namen für vorgestellte Klassen benutzt.

11.3 Bayesian Filtering

Die Ergebnisse des Filterns nur mit bestimmten Regeln liefern nicht optimale Ergebnisse (Spam-Fehler-Rate liegt bei 20%). Es war ein Hinweis, dass die Ergebnisse des Filterns weit von den möglichen Ergebnissen entfernt sind. Mit der Bayes Formel werden Wahrscheinlichkeiten für einzahle Wörter innerhalb jeder E-Mail berechnet, um sicherzustellen, wie oft sie überhaupt auftauchen und wie groß die Wahrscheinlichkeit ist, dass eine beliebige E-Mail eine Spam ist und die Wahrscheinlichkeiten für die Wörter innerhalb eines E-Mails, die die E-Mail zu Ham oder zu Spam klassifizieren. Die vorgeschlagene Anzahl der Wörter anhand deren die gefilterte E-Mail untersucht wird ist auf 15 reduziert. Mit den fünfzehn Wörtern funktioniert das Verfahren sehr gut. Vielleicht lässt sich die Wahrscheinlichkeit besser berechnen. Wichtig ist nach Wörtern zu suchen, deren Wahrscheinlichkeit möglich weit von 0.5 liegt. Auf jeden Fall würde es keinen Sinn machen eine E-Mail für alle Wörter zu testen. Der UBE-Urheber könnte einfach zu einer sehr kurzer Spam einen langen Text einfügen, der nichts mit dem Inhalt zu tun hätte. Der eingefügte Text könnte allerdings den Spamfilter täuschen. Eine zu kleine Anzahl der zu testenden Wörter hätte eine zu große Anzahl von falsch klassifiziertem Ham produziert. Dieser Vorgang wäre für die Benutzer noch schädlicher. Eine große Entscheidung bei dem Filtern mit Bayes spielt die richtige Wahl des Schwellenwerts *threshold*. Mit dem Justieren des Schwellenwerts kann die Genauigkeit des Verfahrens optimiert werden. Eine entscheidende Rolle spielt auch das Trainieren des Filters am Benutzer spezifischen Satz von E-Mails. Das System muss auf einer großer Anzahl von Spam und Ham trainiert werden, um gute Ergebnisse des Filterns zu bekommen. Der Autor des Vorgehens behauptet, dass sein Verfahren mit der Genauigkeit von 99.5% die Spam filtert und dabei nur 0.03% Ham als Spam klassifiziert [2] [3].

11.4 SpamAssassin

SpamAssassin gehört zu den besten Spamfilterungssystemen. Es ist ein *open-source* hybrid Spamfilter, der aus Beyeschen Lerner und einem Satz von 500 heuristischen Regeln besteht, die spezifische Felder nach bestimmten regulären Ausdrücken im E-Mail-Header und im E-Mail-Body suchen. Jeder in einer E-Mail gefundene reguläre Ausdruck wird mit einer bestimmten Anzahl von Punkten bestimmt. Mit dem gewählten Schwellenwert wird die Summe aller in einer E-Mail gesammelten Punkte verglichen. Anhand des Bayeschen Lerner und der Punkte wird entschieden, ob eine E-Mail eine Spam oder ein Ham ist. Der SpamAssassin kann je nach Wahl des Schwellenwerts adjustiert werden, um die Optimalität zu erreichen. Wenn dem Benutzer wichtiger ist den kleineren *fn* zu bekomme, wird der Schwellenwert absichtlich hoch gewählt. Dafür muss er allerdings mit einer höheren Rate von *fp* rechen.
Um zu erfahren wie effektiv der jeweilige Spamfilter ist, wurde *Total Cost Ratio* (TCR) eingeführt. Alle Ham bekommen bestimmten Kostenmaß, je nach Wichtigkeit und je nach Art einer E-Mail. Alle persönlichen E-Mails wurden mit den Kosten 1000 pro Einheit assoziiert. Die geschäftlichen E-Mails wurden mit den Kosten 500 aufgelistet, die E-Commerce mit 100, E-Mailinglisten und Diskussionen mit 50 und die Werbe-E-Mails mit den Kosten 25. Jede Spam die per Hand aus dem Postfach entfernt werden musste wurde mit den Kosten 1 assoziiert. Mit der folgenden Formel $TCR = \frac{fp+tn}{\sum_{x \in fn} C(x)+fp}$ wird überprüft, ob dar Spamfilter besser als manuelles Sortieren funktioniert. Wenn TCR > 1 aus der Formel kommt, heißt es, dass mit der Rücksicht auf Kosten der Spamfilter besser als manuelles Sortieren funktioniert. Je größer der TCR, desto besser für das System.
Die durchgeführte Studie [5] hat gezeigt, dass bei der Verteilung 94% Spam zu 6% Ham der Mensch tendiert 0.16% seiner Ham falsch zu klassifizieren bzw. zu übersehen. Die Fehlerrate des Menschen dient als erstrebenswert für den guten Spamfilter. Es wird auch erwartet, dass ein guter Spamfilter Spam/Ham im Verhältnis 1:1 erreicht. Es heißt, dass die $fp < 5\%$ schon akzeptabel ist [5, SpamAssassin].

11.5 Test

Der Autor führt einige Tests mit dem normalen SpamAssassin (SA), mit dem um einen neuen beyeschen Modell erweiterten SA und noch mit vier unterschiedlichen Klassifikatoren durch. Dazu benutzt er drei unterschiedlich grosse *ham data sets*. Alle Tests werden mit *two-fold crossvalidation* (CV) durchgeführt. Um es zu bewerkstelligen wurden alle drei *ham data sets* jeweils in zwei Subsets nahe 1:1 geteilt. Zuerst wurde ein Subset trainiert und das zweite Subset wurde getestet. Danach wurden die Subsets vertauscht und

das gleiche wurde an vertauschten Subsets für alle drei *ham data sets* durchgeführt. Der Vorteil von CV gegenüber der normalen CV liegt darin, dass man nach jeder Runde die Fehler verifizieren kann.
Die Ergebnisse von SpamAssassin mit dem neuen beyeschen Modell sind deutlich besser als mit dem vorgegebenen SA. Der MLR Klassifikator zeigt noch bessere Ergebnisse und wird als Inspiration für neue Modelle gewählt. Sieben neue Modelle werden entwickelt, um mehrere Vorgänge von Tests zu ermöglichen und sie besser untersuchen zu können. Die besten sind schon inzwischen besser als MLR geworden. Die *ham error rates* der besten Modelle sind so gut wie ein manuelles Sortieren und um Faktor vier besser als der SpamAssassin mit dem neuen Beyeschen Modell. Es wird auch gezeigt, dass es zwecklos war, den Schwellenwert von Wert 0.5 auf 1.0 zu ändern. Man konnte ham_e (ham error rate) auf Null reduzieren, aber dadurch wurde $spam_e$(spam error rate) erheblich erhöht. Nicht zu vergessen ist, dass nicht nur bei wichtigeren Modellen aber auch bei SA die TCR den Wert 1 übersteigt.

12 Conclusion

Ein gut trainiertes beyesches Modell ist ein wesentlicher Bestandteil eines guten Spamfilters. Allein gut gewählte Heuristiken sollten durch ein gutes beyesches Modell ergänzt werden. Man konnte noch die Performance eines Spamfilters mit Suchen nach oft benutzten Phrasen oder Wortpaaren erhöhen. In vielen Fällen hätte dieses Vorgehen die sonst benutzten Wahrscheinlichkeiten besser ausgewählt. Beim Konstruieren eines guten Spamfilters sollte man auf keinen Fall vergessen, den Header einer E-Mail zu überprüfen. In vielen Fällen genügt es den Header zu filtern um zu erfahren, zu welcher Klasse eine E-Mail klassifiziert wird. Obwohl der Schwellenwert nach einer bestimmten Anzahl der durchgeführten Tests vom Spamfilter optimiert wird, sollte zum Schluss dem Benutzter überlassen werden, den Schwellenwert nach seiner Präferenz zu wählen.

Literatur

[1] D. E. Comer. Internetworking with TCP/IP 5th Edition.

[2] P. Graham. www.paulgraham.com/spam.html/. In *A Plan for Spam*, 2002.

[3] P. Graham. www.paulgraham.com/better.html/. In *Better Bayesian Filtering*, 2003.

[4] P. Resnick. Request for Comments: 2822:. In *RFC2822*, 2001.

[5] A. K. Seewald. Combining bayesian and rule score learning:. In *Automated Tuning for SpamAssassin*, 2004.

[6] TekGuard. E-Mail Tutorial.