GI-Fachgruppe
Informatische Bildung in NRW

Informatische Bildung in NRW

Unterrichtsmaterialien für die Sekundarstufe II

Johannes Pieper

Protokolle in Netzwerken

Time, E-Mail und ein selbsterstelltes Protokoll

Autoren

Dieses Dokument ist ein sogenanntes Modul der Materialsammlung, die von der Fachgruppe Informatische Bildung in NRW der Gesellschaft für Informatik heraus gegeben wird. Verantwortliche Autoren für dieses Dokument: Johannes Pieper

Lizenz

Dieses Dokument steht unter der Creative Commons by-nc-sa-Lizenz. Folglich darf es beliebig kopiert und bearbeitet werden, sofern das Folgeprodukt wiederum unter dieser Lizenz vertrieben wird. Eine kommerzielle Nutzung ist nicht erlaubt. https://creativecommons.org/licenses/by-nc-sa/4.0/deed.de

Verweise auf Material

Dieses Modul enthält Hinweise und Verweise (Links) auf Material.Wird eine Seite des Bezugsmaterials dargestellt, ist dieses immer die erste, auch wenn das entsprechende Material aus mehreren besteht. Die Weltkugel verweist dabei auf die entsprechende Webseite und das Dokumentensymbol hat einen relativen Pfad hinterlegt, der die Nutzung der Verzeichnisstruktur der Materialsammlung voraussetzt.

Kontakt

Ihre Ansprechpartner für die vorliegenden Materialien sind Daniel Losch daniel.losch@gi.de und Johannes Pieper pieper@informatiktag-nrw.de

protokolle.tex

git
Version: dd1fc2
Stand: 26. Juni 2026 18:18
Zuletzt bearbeitet von: Johannes Pieper

1 Beschreibung

In der Oberstufe ist das Thema Netzwerke im Kernlehrplan des Landes Nordrhein-Westfalen aufgeführt. In diesem Modul liegt dabei der Schwerpunkt auf Protokollen und ihrem Aufbau. Die weiteren Elemente von Netzwerken sind zum größten Teil in Form von Referaten durch die Schülerinnen und Schüler geplant. Die Themen dieser Referate werden zu Beginn des Moduls teilweise mit den Lernenden erarbeitet und verteilt, so dass sie zu passenden Zeitpunkten gehalten werden können.

Begonnen wird bei den Protokollen mit einem einfachen Time-Protokoll, bei dem die Grundlagen für den Aufbau von Protokollen erarbeitet wird. Darauf aufbauend wird mit E-Mails gearbeitet. Dabei wird das POP3-Protokoll praktisch eingesetzt um per Hand E-Mails zu empfangen und genauso mit dem SMTP-Protokoll E-Mails versandt.

Den Abschluss des Moduls bildet die Erarbeitung eines eigenen Protokolls für z. B. die Kommunikation, wie sie aus verschiedenen Messengerdiensten bekannt ist. Diese Erarbeitung erfolgt in Gruppen und wird zum Abschluss von anderen Gruppen erprobt. Eine Implementierung ist dabei nicht vorgesehen.

Der hier vorgestellte Ablauf basiert auf einer in der Jahrgangsstufe Q2 durchgeführten Reihe.

2 Ablauf und Material

Zu Beginn des Moduls wird bei den Schülerinnen und Schülern im Plenum das Vorwissen zu Netzwerken abgefragt. Dabei werden die genannten Begriffe zentral gesammelt. Dabei können thematisch zusammenhängende Begriffe bereits örtlich nahe beieinander geschrieben werden. Aus diesen Begriffen und Begriffsgruppen werden dann Themen für Referate im Umfang zwischen 10 und 20 Minuten, je nach Themengebiet. Die folgenden Begriffe sollten dabei auf jeden Fall mit aufgetragen werden und deshalb ggf. von der Lehrkraft vor der Verteilung ergänzt werden:

- Client/Server und Peer to Peer
- IPv4 und IPv6
- UDP und TCP
- DNS
- DHCP
- OSI-Schichtenmodell
- Router (incl. Netzwerkmaste und Gateway)
- Topologien
- Firewall

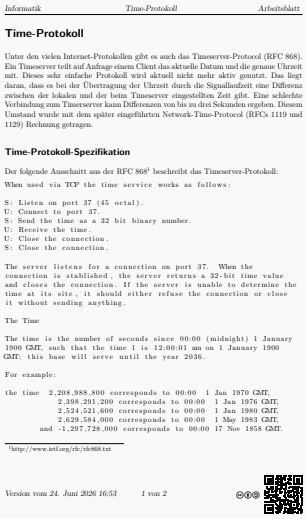
Als Einstiegsprotokoll wird das veraltete Timeprotokoll nach dem RFC 868 genutzt. Das Arbeitsblatt dazu enthält eine einfache Beschreibung des Protokolls mit dem Ablauf der Kommunikation. Zur Durchführung wird das Programm Telnet auf den Rechnern der Schüler benötigt. Ein entsprechendes Serverprogramm ist mitgeliefert. Dieses lässt

sich nur mit administrativen Rechten so starten, dass es auf dem Orgnialport lauscht. Ist dieses nicht möglich, kann auch ein freier Port wie 10037 im Programmtext angegeben werden.

Für die Implementierung in einem Grundkurs in NRW muss beachtet werden, dass nur die Materialien für den Leistungskurs passende Klassen bereit stellen und an diesen muss eine kleine Änderung vorgenommen werden: An einer Stelle muss der Hinweis bzgl. des Zeichensatz beachtet werden.

Material

- ab_time_protokoll.pdf



Informatik *Time-Protokoll* *Arbeitsblatt*

Time-Protokoll

Unter den vielen Internet-Protokollen gibt es auch das Timeserver-Protocol (RFC 868). Ein Timeserver teilt auf Anfrage einem Client das aktuelle Datum und die genaue Uhrzeit mit. Dieses sehr einfache Protokoll wird aktuell nicht mehr aktiv genutzt. Das liegt daran, dass es bei der Übertragung der Uhrzeit durch die Signallaufzeit eine Differenz zwischen der lokalen und der beim Timeserver eingestellten Zeit gibt. Eine schlechte Verbindung zum Timerserver kann Differenzen von bis zu drei Sekunden ergeben. Diesem Umstand wurde mit dem später eingeführten Network-Time-Protocol (RFCs 1119 und 1129) Rechnung getragen.

Time-Protokoll-Spezifikation

Der folgende Ausschnitt aus der RFC 868[1] beschreibt das Timeserver-Protokoll:

When used via TCP the time service works as follows:

S: Listen on port 37 (45 octal).
U: Connect to port 37.
S: Send the time as a 32 bit binary number.
U: Receive the time.
U: Close the connection.
S: Close the connection.

The server listens for a connection on port 37. When the connection is stablished, the server returns a 32-bit time value and closes the connection. If the server is unable to determine the time at its site, it should either refuse the connection or close it without sending anything.

The Time

The time is the number of seconds since 00:00 (midnight) 1 January 1900 GMT, such that the time 1 is 12:00:01 am on 1 January 1900 GMT; this base will serve until the year 2036.

For example:

the time 2,208,988,800 corresponds to 00:00 1 Jan 1970 GMT;
2,398,291,200 corresponds to 00:00 1 Jan 1976 GMT;
2,524,521,600 corresponds to 00:00 1 Jan 1980 GMT;
2,629,584,000 corresponds to 00:00 1 May 1983 GMT;
and -1,297,728,000 corresponds to 00:00 17 Nov 1858 GMT.

[1] http://www.ietf.org/rfc/rfc868.txt

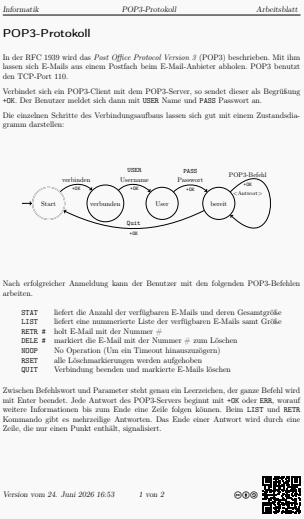
Version vom 24. Juni 2026 16:53 *1 von 2*

Mit POP3 und SMTP stehen zwei weitere Protokolle zur Verfügung, die sich auch gut „von Hand“ bedienen lassen. Da heutzutage aber fast alle gängigen E-Mail-Server nur noch verschlüsselte Verbindungen zulassen muss auf einen eigenen Zugegriffen werden. Dieser hat auch noch den Vorteil, dass er so eingestellt werden kann, dass für SMTP keine Authentifizierung nötig ist. Diese ist laut Protokoll auch nicht nötig.

Mit dem Installationsskript lässt sich ein entsprechender E-Mail-Server direkt auf einem Raspberry Pi aufsetzen. Bei diesem muss aus Sicherheitsgründen sichergestellt werden, dass er nur aus dem lokalen Netzwerk in dem die Lernenden arbeiten erreichbar ist. Gleichzeitig ist der Server so konfiguriert, dass er keine E-Mails nach außen weiterreicht.

Material

- ab_pop3_protokoll.pdf



Informatik *POP3-Protokoll* *Arbeitsblatt*

POP3-Protokoll

In der RFC 1939 wird das *Post Office Protocol Version 3* (POP3) beschrieben. Mit ihm lassen sich E-Mails aus einem Postfach beim E-Mail-Anbieter abholen. POP3 benutzt den TCP-Port 110.

Verbindet sich ein POP3-Client mit dem POP3-Server, so sendet dieser als Begrüßung `+OK`. Der Benutzer meldet sich dann mit `USER` Name und `PASS` Passwort an.

Die einzelnen Schritte des Verbindungsaufbaus lassen sich gut mit einem Zustandsdiagramm darstellen:

Nach erfolgreicher Anmeldung kann der Benutzer mit den folgenden POP3-Befehlen arbeiten.

`STAT`	liefert die Anzahl der verfügbaren E-Mails und deren Gesamtgröße
`LIST`	liefert eine nummerierte Liste der verfügbaren E-Mails samt Größe
`RETR #`	holt E-Mail mit der Nummer #
`DELE #`	markiert die E-Mail mit der Nummer # zum Löschen
`NOOP`	No Operation (Um ein Timeout hinauszuzögern)
`RSET`	alle Löschmarkierungen werden aufgehoben
`QUIT`	Verbindung beenden und markierte E-Mails löschen

Zwischen Befehlswort und Parameter steht genau ein Leerzeichen, der ganze Befehl wird mit Enter beendet. Jede Antwort des POP3-Servers beginnt mit `+OK` oder `ERR`, worauf weitere Informationen bis zum Ende eine Zeile folgen können. Beim `LIST` und `RETR` Kommando gibt es mehrzeilige Antworten. Das Ende einer Antwort wird durch eine Zeile, die nur einen Punkt enthält, signalisiert.

Version vom 24. Juni 2026 16:53 *1 von 2*

- ab_smtp_protokoll.pdf



Informatik *SMTP-Protokoll* *Arbeitsblatt*

SMTP-Protokoll

Das SMTP-Protokoll (Simple Mail Transfer Protocol) wird in der RFC 821 und RFC 822 beschrieben. Es beschreibt, wie der Versand von E-Mails zu erfolgen hat. Dabei ist nicht nur der Versand an einen Mail-Server abgedeckt, sondern auch zwischen verschiedenen Mail-Servern. Diese dienen als Zwischenstationen, bis eine Mail das Zielpostfach erreicht hat. Das SMTP-Protokoll nutzt standardmäßig den TCP Port 25.

Anmeldung am SMTP-Server

Das SMTP-Protokoll beinhaltet keine Authentifizierungsmechanismen. Daher ist es möglich, dass jeder beliebige Benutzer Mails über jeden SMTP-Server versenden kann. Um dieses zu verhindern, wurde eine Erweiterung des SMTP-Protokolls mit SMTP-Auth (Simple Mail Transfer Protocol Service Extension for Authentication) eingeführt. Bei diesem Verfahren müssen zunächst ID und Passwort entsprechend kodiert an den SMTP-Server übermittelt werden, bevor eine Mail versandt werden kann.

SMTP-Befehle

`HELO`	(Hello) Startet die SMTP Sitzung und identifiziert den Client am Server.
`EHLO`	(Extended Hello) Liefert nach dem Start zusätzliche Informationen über den Server zurück.
`MAIL FROM:`	Leitet die Mailübertragung ein und liefert gleich die Absender-Adresse mit.
`RCPT TO:`	Gibt die Adresse eines oder mehrerer Empfänger an; das Kommando kann deshalb mehrmals ausgeführt werden.
`DATA`	Kennzeichnet den Beginn der eigentlichen E-Mail-Nachricht; das Ende wird mit einem einzelnen Punkt in einer Zeile gekennzeichnet.
`RSET`	Bricht eine bereits eingeleitete Mailübertragung ab, die Verbindung zwischen Client und Server bleibt bestehen.
`VRFY`	(Verify) überprüft, ob eine Empfänger-Adresse verfügbar ist.
`NOOP`	Bewirkt eine Antwort vom Server; verhindert die Trennung der Verbindung durch einen Timeout.
`QUIT`	Beendet die Verbindung zum SMTP Server; der Server liefert eine letzte Antwort zurück.

Reihenfolge und Pflichtfelder

Beim Mailversand über SMTP ist zu beachten, dass eine Reihenfolge der Befehle einzuhalten ist. So wird zum Einleiten einer Mail mit `MAIL FROM:` begonnen, es folgt ein oder

Version vom 24. Juni 2026 16:53 *1 von 2*

Auf Grundlage der Erfahrungen können die Lernenden über die Zusammenhänge zwischen dem SMTP-Protokoll und dem Versandt von Spam-Mails diskutieren. Bei dieser Diskussion sollten sowohl die Grundlagen für den Spam-Versand als auch das Aufzeigen von Möglichkeiten, wie das Protokoll erweitert bzw. verbesserte werden sollte, beleuchtet werden.

Time-Protokoll

Unter den vielen Internet-Protokollen gibt es auch das Timeserver-Protocol (RFC 868). Ein Timeserver teilt auf Anfrage einem Client das aktuelle Datum und die genaue Uhrzeit mit. Dieses sehr einfache Protokoll wird aktuell nicht mehr aktiv genutzt. Das liegt daran, dass es bei der Übertragung der Uhrzeit durch die Signallaufzeit eine Differenz zwischen der lokalen und der beim Timeserver eingestellten Zeit gibt. Eine schlechte Verbindung zum Timerserver kann Differenzen von bis zu drei Sekunden ergeben. Diesem Umstand wurde mit dem später eingeführten Network-Time-Protocol (RFCs 1119 und 1129) Rechnung getragen.

Time-Protokoll-Spezifikation

Der folgende Ausschnitt aus der RFC 868[1] beschreibt das Timeserver-Protokoll:

```
When used via TCP the time service works as follows:

S: Listen on port 37 (45 octal).
U: Connect to port 37.
S: Send the time as a 32 bit binary number.
U: Receive the time.
U: Close the connection.
S: Close the connection.

The server listens for a connection on port 37.  When the
connection is stablished, the server returns a 32-bit time value
and closes the connection. If the server is unable to determine the
time at its site, it should either refuse the connection or close
it without sending anything.

The Time

The time is the number of seconds since 00:00 (midnight) 1 January
1900 GMT, such that the time 1 is 12:00:01 am on 1 January 1900
GMT; this base will serve until the year 2036.

For example:

the time  2,208,988,800 corresponds to 00:00  1 Jan 1970 GMT,
            2,398,291,200 corresponds to 00:00  1 Jan 1976 GMT,
            2,524,521,600 corresponds to 00:00  1 Jan 1980 GMT,
            2,629,584,000 corresponds to 00:00  1 May 1983 GMT,
        and -1,297,728,000 corresponds to 00:00 17 Nov 1858 GMT.
```

[1] http://www.ietf.org/rfc/rfc868.txt

Aufgabe

a) Geben Sie an, aus welchen Elementen das Protokoll besteht.

b) Stellen Sie den möglichen Ablauf einer Kommunikation in einem Sequenzdiagramm dar.

c) Informieren Sie sich über die Bedeutung der RFC (Requests for Comments)

d) Verbinden Sie sich mithilfe des Programms telnet in der Kommandozeile mehrfach mit einem Timeserver und interpretieren Sie seine Ausgabe. Die Adresse eines entsprechenden Servers erfahren Sie von Ihrem Lehrer.

e) Schreiben Sie in einer Gruppe einen Timeclient, der die aktuelle Uhrzeit von einem Timeserver holt und sie anzeigt. Beachten Sie dabei die unteren Hinweise.

Hinweise zu Java:

1. Für die Umwandlung der vom Client empfangenen Daten im String s können folgende Anweisungen verwendet werden:

```
long ltime = 0;
for (int i=0; i<4;i++) {
    ltime = ltime * 256 + (int)s.charAt(i);
}
```

2. Da Java standardmäßig vom 1.1.1970 aus rechnet, muss von der berechneten Zeit etwas abgezogen werden:

```
long groundtime = 2208988800L;
```

3. Die Ausgabe der Zeit in der Variable date funktioniert mit der Methode:

```
SimpleDateFormat date = new SimpleDateFormat("MM/dd/yyyy␣
   HH:mm:ss␣z");
date.setTimeZone(TimeZone.getTimeZone("UTC"));
String sdate = date.format (new java.util.Date(ltime *1000));
```

   Dazu müssen Sie java.text.* importieren.

4. Beim Einlesen der Daten muss der InputStream auf den passenden Zeichensatz gestellt werden, da Java standardmäßig immer zwei Byte gleichzeitig einliest.

```
new InputStreamReader(socket.getInputStream(),"ISO-8859-1");
```

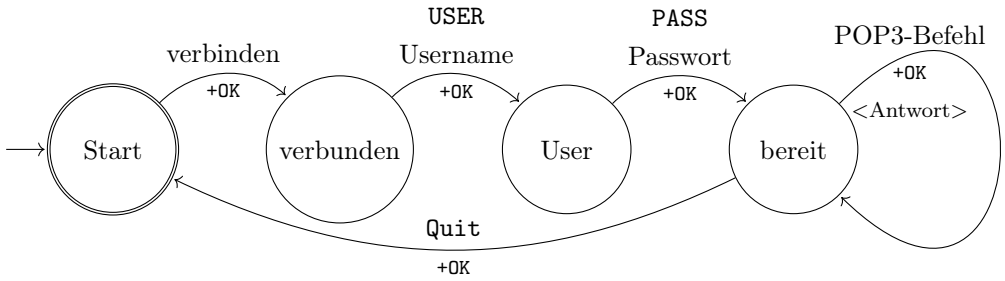
   Bei der Verwendung der Abiturklassen aus NRW, muss diese Änderung in den Klassen direkt vorgenommen werden.

POP3-Protokoll

In der RFC 1939 wird das *Post Office Protocol Version 3* (POP3) beschrieben. Mit ihm lassen sich E-Mails aus einem Postfach beim E-Mail-Anbieter abholen. POP3 benutzt den TCP-Port 110.

Verbindet sich ein POP3-Client mit dem POP3-Server, so sendet dieser als Begrüßung `+OK`. Der Benutzer meldet sich dann mit `USER` Name und `PASS` Passwort an.

Die einzelnen Schritte des Verbindungsaufbaus lassen sich gut mit einem Zustandsdiagramm darstellen:



Nach erfolgreicher Anmeldung kann der Benutzer mit den folgenden POP3-Befehlen arbeiten.

`STAT`	liefert die Anzahl der verfügbaren E-Mails und deren Gesamtgröße
`LIST`	liefert eine nummerierte Liste der verfügbaren E-Mails samt Größe
`RETR #`	holt E-Mail mit der Nummer #
`DELE #`	markiert die E-Mail mit der Nummer # zum Löschen
`NOOP`	No Operation (Um ein Timeout hinauszuzögern)
`RSET`	alle Löschmarkierungen werden aufgehoben
`QUIT`	Verbindung beenden und markierte E-Mails löschen

Zwischen Befehlswort und Parameter steht genau ein Leerzeichen, der ganze Befehl wird mit Enter beendet. Jede Antwort des POP3-Servers beginnt mit `+OK` oder `ERR`, worauf weitere Informationen bis zum Ende eine Zeile folgen können. Beim `LIST` und `RETR` Kommando gibt es mehrzeilige Antworten. Das Ende einer Antwort wird durch eine Zeile, die nur einen Punkt enthält, signalisiert.

Anmerkung: Im POP3 Protokoll ist vermerkt, dass die zum Löschen markierten Mails erst mit dem Schließen der Verbindung gelöscht werden. Außerdem dient der Befehl `NOOP` dazu, die Verbindung aufrecht zu erhalten, so dass es zu keinem Timeout kommt.

Aufgabe

a) Erstellen Sie ein Sequenzdiagramm, indem die Anmeldung an einen POP3-Server, das Holen der ersten Nachricht und das Schließen der Verbindung dargestellt wird.

b) Nutzen Sie in der Kommandozeile das Programm Telnet, um sich mit einem POP3-Server zu verbinden. Die genaue Adresse und die Zugangsdaten erfahren Sie von ihrem Lehrer.

c) Führen Sie dort folgende Aktionen aus.

- Lassen Sie sich die Liste der vorhandenen E-Mails geben.
- Lesen Sie eine der vorhandenen Mails.
- Löschen Sie eine der Mails.

d) Überprüfen Sie das Sequenzdiagramm auf seine Korrektheit anhand Ihrer Aktionen mit Telnet.

e) Bei Übertragung mithilfe des POP3-Protokolls werden die E-Mails so übertragen, wie sie auf dem Server und Client gespeichert sind. Schreiben Sie das Schema des Aufbaus einer E-Mail in diesem Format auf.

SMTP-Protokoll

Das SMTP-Protokoll (Simple Mail Transfer Protocol) wird in der RFC 821 und RFC 822 beschrieben. Es beschreibt, wie der Versand von E-Mails zu erfolgen hat. Dabei ist nicht nur der Versand an einen Mail-Server abgedeckt, sondern auch zwischen verschiedenen Mail-Servern. Diese dienen als Zwischenstationen, bis eine Mail das Zielpostfach erreicht hat. Das SMTP-Protokoll nutzt standardmäßig den TCP Port 25.

Anmeldung am SMTP-Server

Das SMTP-Protokoll beinhaltet keine Authentifizierungsmechanismen. Daher ist es möglich, dass jeder beliebige Benutzer Mails über jeden SMTP-Server versenden kann. Um dieses zu verhindern, wurde eine Erweiterung des SMTP-Protokolls mit SMTP-Auth (Simple Mail Transfer Protocol Service Extension for Authentication) eingeführt. Bei diesem Verfahren müssen zunächst ID und Passwort entsprechend kodiert an den SMTP-Server übermittelt werden, bevor eine Mail versandt werden kann.

SMTP-Befehle

`HELO`	(Hello) Startet die SMTP Sitzung und identifiziert den Client am Server.
`EHLO`	(Extended Hello) Liefert nach dem Start zusätzliche Informationen über den Server zurück.
`MAIL FROM:`	Leitet die Mailübertragung ein und liefert gleich die Absender-Adresse mit.
`RCPT TO:`	Gibt die Adresse eines oder mehrerer Empfänger an; das Kommando kann deshalb mehrmals ausgeführt werden.
`DATA`	Kennzeichnet den Beginn der eigentlichen E-Mail-Nachricht; das Ende wird mit einem einzelnen Punkt in einer Zeile gekennzeichnet.
`RSET`	Bricht eine bereits eingeleitete Mailübertragung ab; die Verbindung zwischen Client und Server bleibt bestehen.
`VRFY`	(Verify) überprüft, ob eine Empfänger-Adresse verfügbar ist.
`NOOP`	Bewirkt eine Antwort vom Server; verhindert die Trennung der Verbindung durch einen Timeout.
`QUIT`	Beendet die Verbindung zum SMTP Server; der Server liefert eine letzte Antwort zurück.

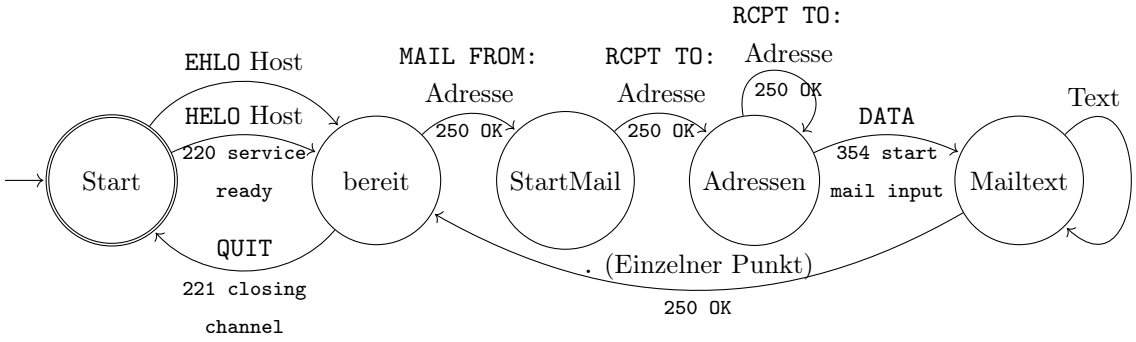
Reihenfolge und Pflichtfelder

Beim Mailversand über SMTP ist zu beachten, dass eine Reihenfolge der Befehle einzuhalten ist. So wird zum Einleiten einer Mail mit `MAIL FROM:` begonnen, es folgt ein oder

mehrmals die Angabe der Empfänger mit `RCPT TO:` um anschließend mit `DATA` die Mail einzuleiten. Der Server antwortet dann mit entsprechenden Statuscodes.

Innerhalb der folgenden Datenübertragung kommt zuerst der Kopf einer Mail. Jeweils in einer Zeile sind dazu verschiedene Angaben zu machen. Mit `from:` und `to:` werden der Absender und der oder die Empfänger angegeben. Da diese in der Regel nicht weiter verarbeitet werden, können sie sich von den Angaben zu den Befehlen des Protokolls unterscheiden. Durch `subject:` kann man den Betreff der Mail angeben.

Abgeschlossen wird eine Mail mit einer Zeile, die nur einen Punkt als einziges Zeichen enthält. Damit wird der Datenblock abgeschlossen und die Mail versandt.



Aufgabe

a) Erstellen Sie ein Sequenzdiagramm, das den Ablauf beschreibt, wenn Sie an eine Freundin eine Mail versenden, die nur aus einer Begrüßung, einem kurzen Text und einer Verabschiedung besteht.

b) Nutzen Sie in der Kommandozeile das Programm Telnet, um sich mit einem SMTP-Server zu verbinden. Die genaue Adresse und die Zugangsdaten erfahren Sie von ihrem Lehrer.

c) Geben Sie an, wie mit dem Protokoll die Blindkopie (BCC) realisiert wird.

d) Beschreiben Sie die Auswirkungen des SMTP-Protokolls bzgl. der Anmeldung am SMTP-Server für den heutigen Datenverkehr. Gehen Sie dabei besonders auf Möglichkeiten des Missbrauchs ein.