ZCONNECT[R]

Das Datenaustauschformat für MailBox-Netze

Version 3.1

Wolfgang Mexner, Felix Heine, Matthias Jung,
Hartmut Schröder, Martin Husemann,
Rena Tangens & padeluun

März 1995

Copyright und Warenzeichen

ZERBERUS GmbH, Friedland:
ZCONNECT Datenaustauschformat für MailBox-Netzwerke, Bielefeld:
Verlag Art d'Ameublement, 1994, ISBN 3-9802182-3-6

Verlag	:	Art d'Ameublement, Bielefeld
Satz	:	TEX
Belichtung	:	HP LaserJet III auf Folie
Druck	:	OFF//SET Druckerei im Umweltzentrum, Bielefeld

Eingetragene Warenzeichen:

Die in den Dokumentationen und im Programm genannten Programm- und Verfahrensnamen sind möglicherweise urheberrechtlich und/oder warenzeichenrechtlich geschützt. Fehlende Hinweise auf bestehende Urheberrechte oder Warenzeichen berechtigen nicht zu der Annahme, daß ein solches nicht existiert.

- ZERBERUS ist ein eingetragenes Warenzeichen der ZERBERUS GmbH, Deutschland
- ZCONNECT ist ein eingetragenes Warenzeichen der ZERBERUS GmbH, Deutschland
- ZMODEM ist ein eingetragenes Warenzeichen der Omen Technology, Inc., USA
- PKZIP und PKUNZIP sind eingetragene Warenzeichen der PKWARE, Inc., USA
- IBM, PS/2 und OS/2 sind eingetragene Warenzeichen der International Business Machines Corporation, USA
- MS-DOS, GW-BASIC, MS-WINDOWS und MS-WORD sind eingetragene Warenzeichen der Microsoft Corporation, USA
- FIDO und FIDONET sind eingetragene Warenzeichen von Tom Jennings, Fido Software, USA
- UNIX ist ein eingetragenes Warenzeichen der AT&T, Inc. und USL, Inc.
- RSA ist ein eingetragenes Warenzeichen der RSA Data Security, Inc.
- PGP ist ein eingetragenes Warenzeichen von Phil Zimmermann.

Inhaltsverzeichnis

Kapitel I: Einleitung

Der ZCONNECT-Standard beschreibt den Datenaustausch zwischen verschiedenen Systemen in einem MailBox-Netzwerk. Die Verfahren wurden entworfen und ausgewählt mit Blick auf mögliche Erweiterbarkeit ohne Änderungen in der darauf basierenden Software und der möglichst einfachen Konvertierbarkeit in andere Datenformate. Hierbei wurden speziell das alte Z-NETZ Format und das InterNet / UseNet Datenformat (s. RFC821 / 822 / 1036) berücksichtigt.

Ergebnis ist nun ein hinreichend einfaches, aber gleichzeitig sehr mächtiges Verfahren. Da ZCONNECT auf den Erfahrungen der genannten Protokolle aufsetzt, aber komplett neu entworfen wurde, konnte eine sehr viel kompaktere und schlichtere Struktur bei gleicher Leistungsfähigkeit erreicht werden.

Wir beschreiben die unterschiedlichen Protokollebenen in drei Kapiteln. Zunächst wird die eigentliche Datenübertragung (Online-Phase) erläutert. Im zweiten Kapitel stellen wir die übertragenen Daten und damit das Nachrichtenformat vor. Schließlich finden Sie im Anhang eine Beschreibung eines Befehlsvorrats eines MAPS-Roboters, der automatisch Brettbestellungen für an MailBox-Systeme angeschlossene Points durchführt.

Hinweis: In der deutschen Sprache gibt es leider noch keinen neutralen Ausdruck für Personen, der beide Geschlechter umfaßt. Da wir den Text nicht durch beständige Nennung beider Formen unnötig kompliziert machen wollten, verwenden wir in dieser Dokumentation ausschließlich die weibliche Form. Selbstverständlich sind mit den Bezeichnungen Absenderin, Empfängerin, Systembetreiberin etc. die entsprechenden männlichen Personen mitgemeint.

I.1 Danksagung

Wir danken dem FoeBuD e.V. für ein Dauertestsystem, Peter Mandrella, Marc Zimmermann und allen anderen Point-Programmiererinnen sowie Frank Scheizel für viele Tips und Anregungen, Christian Mock für die Zusendung der Zeitzonentabelle (zusammengestellt von Gary Dixon), Hans-Christian Fricke, Garry Glendown, Michael Grube, Toni Guenzel-Peltner, Joachim Haas, Daniel Kroening, Holger Lembke, Dirk Meyer, Helmut Neumann, Sandro Paolino, Goetz Schuchard, Ralph Seichter, Christoph Teuber, Oliver Wagner, Matthias Watermann und Andreas Wittkemper für die Mitarbeit im ZCONNECT-Wahlgremium, Hinrich Donner für die Moderation der ZCONNECT-Wahlen, Andrea Hildebrand und allen anderen, die uns bei diesem harten Stück Arbeit unterstützt haben.

Insbesondere gilt unser Dank Hartmut Schröder und Martin Husemann, ohne die es ZCONNECT nie gegeben hätte.

I.2 Copyright

Die Dokumentation kann im Buchhandel unter der internationalen Bestellnummer ISBN 3-9802182-3-6 oder direkt bei der ZERBERUS GmbH bestellt werden (Vorkasse, Scheck oder bar). Die Doku ist im Format DIN A5, gebunden, mit einer Diskette versehen und kostet 30 DM.

Kapitel II: Die Datenübertragung

II.1 ZCONNECT implementieren

Sie können seit etwa Mitte 1993 einen Auszug aus einem C-Source kostenlos erhalten, der das hier beschriebene Protokoll implementiert. Auf der Basis dieses Quelltextes werden Sie sicher schnell zu einer eigenen Version gelangen und insbesondere die kostenintensive und mühselige Online-Testphase verkürzen.

Aber auch wenn Sie nicht in "C" arbeiten, werden Sie mit wenig Aufwand die Übertragungsphase realisieren können. Sie benötigen dazu:

- Eine 16-bit CRC Routine nach dem kX-Modem Polynom
- Eine Verwaltung für "Header" (siehe unten)
- Dateitransportprotokolle (z.B. Z-MODEM), die Sie aber auch als externe Programme aufrufen können.

In der folgenden Beschreibung werden Texte gelegentlich in C-Notation angegeben. Hier ein kurze Erläuterung:

`\r`	Return, <CR> Dezimal 13, Hex. 0D
`\n`	Newline, <LF>, Dezimal 10, Hex. 0A

Die Anführungsstriche (") gehören nicht zu den Texten, sondern dienen nur deren Begrenzung.

II.2 Gesamtablauf

Grobskizze eines kompletten Verbindungsaufbaus:

1. Login (einmalig)
2. Austausch der Systeminformationen (einmalig)
3. Austausch von Daten (mehrmals hintereinander)
4. Logoff (einmalig)

Eine etwas feinere Skizze des Verbindungsaufbaus für die Anruferin (ohne Fehlerkorrektur):

1. Vorbereitungsphase (noch offline)
   - Auswahl der anzuwählenden MailBox
   - Falls Daten für die MailBox vorhanden sind, diese packen
   - Verbindung zur anderen MailBox herstellen
2. Loginphase (Online)
   - ← Warten auf Username
   - → Usernamen "`zconnect`" eingeben
   - ← Warten auf Passwort
   - → Passwort "`0zconnec`" eingeben
   - ← Warten auf "`BEGIN`"
3. Austausch der Systeminformationen
   - → Senden der eigenen Systeminformationen
   - ← Empfangen der Systeminformationen des anderen
   - Ermitteln der Gemeinsamkeiten
   - → Senden der endgültigen Verbindungsdaten, die für diese Verbindung gelten

   ← Empfangen der Bestätigung der endgültigen Daten
4. Datenaustausch
   → Senden der Anforderung (z.B. “Mails holen”, “Mails senden” oder “Mails holen und senden”)
   ← Empfangen der Rückmeldung (z.B. Meldung, wieviel kByte an Mails bereitliegen)
   → Aufforderung zum Ausführen des Kommandos oder Senden einer Abbruch-Aufforderung
   ← Empfangen der Bestätigung des Kommandos
   ← Gegebenenfalls warten, bis Mails bereitgestellt sind
   ← Umschalten auf Filetransferprotokoll, Daten holen, zurückschalten auf `ZCONNECT`
5. Weiterer Datenaustausch
   → Senden der nächsten Anforderung (z.B. Mails senden)
   ← Empfangen der Bestätigung
   → Senden, ob Anforderung auch ausgeführt werden soll
   ← Empfangen der Bestätigung der Kommandos
   → Umschalten auf Filetransferprotokoll, Daten senden, zurückschalten auf `ZCONNECT`
6. Logoff
   → Senden der Logoffanforderung
   ← Empfangen der Bestätigung der Logoffanforderung
   → Senden der endgültigen Logoffanforderung
   ← Empfangen Bestätigung Logoff
7. Nacharbeiten (Offline)
   - Merken der neuesten Parameter der Box, die angerufen wurde
   - Files und Mails löschen, die nicht mehr gebraucht werden
   - Verarbeiten (Einsortieren der empfangenen Mails)

Eine etwas feinere Skizze des Anrufes für die angerufene MailBox (ohne Fehlerkorrektur):

1. Loginphase
   ← Verbindung wird hergestellt (Online)
   → Loginmeldung senden (kurz, da sie nicht abgebrochen wird!)
   → Senden “`Username:`”
   ← Empfangen des Usernamens “`zconnect`”
   → Senden “`Passwort:`”
   ← Empfangen des Passwortes “`0zconnec`”
   - `ZCONNECT` Übertragungsmodul aktivieren

   Das `ZCONNECT`-Übertragungsmodul macht nun folgendes:
   → Senden drei mal “`BEGIN\r\n`” mit je 0,5 Sekunden Pause. Nach dem letzten “`BEGIN`” erfolgt eine Pause von 1 Sekunde. Die Zeitspanne von “`Username:`” bis zum letzten “`BEGIN`” darf maximal 2 Minuten betragen.
2. Austausch Systeminformationen
   ← Empfangen der Systeminformationen des anderen
   → Senden der eigenen Systeminformationen
   ← Empfangen der endgültigen Verbindungsdaten
   → Senden der Empfangsbestätigung
3. Datenaustausch

← Empfangen der Anforderung (z.B. welche Mailtypen sollen gesendet oder empfangen werden)

- Überprüfen und schätzen, wie lange das Bereitstellen der Daten dauern wird - diese sollten normalerweise gepackt bereitliegen. Packzeiten treten nur bei einem Umpacken auf einen neuen Packer auf, sowie eventuelle Kopierzeiten.

→ Senden der Bestätigung, welche der gewünschten Informationen verfügbar sind (PUT), wie lange es dauern wird diese bereitzustellen und welche Mailtypen von der Gegenstelle gewünscht sind. Hier kann die angerufene auch bereits mitteilen, daß sie gar nichts hat.

← Empfangen der Bestätigung des Kommandos und der Mitteilung ob die Gegenstelle die Daten nun endgültig haben möchte (`EXECUTE` und `WAIT`), oder ob die Daten (dieses bezieht sich nur auf das aktuelle Mailpaket) z.B. erst nach dem Logoff vorbereitet werden sollen.

→ Senden der Bestätigung

→ Senden “Warten, bis Packen beendet” (optional), sofern eine Wartezeit ¿ 0 Sekunden angegeben wurde.

- Packen der Nachrichten

→ Senden “Packen beendet”, sofern eine Wartezeit ¿ 0 Sekunden angegeben wurde.

→ Umschalten auf Filetransferprotokoll, Daten senden und/oder empfangen, zurückschalten auf `ZCONNECT`

4. Weiterer Datenaustausch und Logoff

   Der weitere Datenaustausch erfolgt wieder analog dem ersten Zyklus. Die Kontrolle behält hierbei immer die Angerufene. Die Angerufene hat nun die Möglichkeit, nur einen Teil der Daten bereitzustellen, oder den Transfer (z.B. die Festplatte ist voll oder das Kommando unbekannt) vollständig abzulehnen. Die Anruferin wird solange Mailpakete von der Gegenstelle anfordern, solange diese mitteilt, daß noch Archive zur Übertragung bereitliegen. Vor jedem Aufruf des Transferprotokolls kann über ’Wait’ eine Wartezeit vereinbart werden. Die Verbindung wird beendet, wenn eine Seite in einem Paket eine Logoff-Aufforderung sendet, was jederzeit erfolgen kann. Der Verbindungsabbruch erfolgt jeweils nach einem vollständigen Zyklus. Ein sofortiger Verbindungsabbruch erfolgt, wenn das aufgerufene Übertragungsprotokoll einen Fehler gemeldet hat. Das zuletzt übertragene Datenpaket wird in diesem Falle als nicht übertragen gewertet.

5. Nacharbeit (Offline)
    - Merken der neuesten Parameter der Box, die angerufen hat.
    - Vorbereiten der Datenarchive, die mit dem Kommando `EXECUTE: L` erst nach dem Logoff bereitgestellt werden sollten.
    - Files und Mails löschen, die nicht mehr gebraucht werden
    - Verarbeiten (Einsortieren der empfangenen Mails)

II.3 Login

Wir haben das Login-Verfahren festgeschrieben[1], damit alle ZCONNECT-MailBoxen weltweit Datenaustausch selbständig durchführen können - auch wenn die Systeme noch nie vorher miteinander kommuniziert haben.

Dabei geht das Login-Verfahren von drei Bedingungen aus:

1. Der Einlogname ist immer identisch, ebenso das Passwort. Dadurch ist es möglich, ZCONNECT als Benutzerin in ein System einzutragen, das ansonsten keinen Eingriff in die Einlogprozedur erlaubt (z.B. VMS oder UNIX), dann aber, sobald der allgemeine ZCONNECT-Login gelungen ist, nochmal Systemname und Passwort abzufragen.
2. Manche Systeme erlauben kein Login ohne Passwortabfrage oder erzwingen bestimmte (kryptische) Passworte. Wir haben daher Login *und* Passwort definiert, nicht wie in alten Verfahren nur das Login.
3. Nicht immer kann die Formulierung der Login- oder Passwortabfrage konfiguriert werden, auch ein Abbruch der Titelmeldung vor dem Login ist nicht systemunabhängig definierbar. Daher hat die Anruferin auf eine Vielzahl von Schlüsselwörtern zu reagieren und anschließend noch eine Pause abzuwarten (damit das Schlüsselwort zur Not auch in die Titelmeldung geschrieben werden kann).

Die Senderin wartet solange, bis die Empfängerin in ihrer Loginmeldung einen der folgenden Strings sendet: "`ogin`", "`OGIN`", "`ame`", "`AME`" und eine Übertragungspause von einer Sekunde gefolgt ist. Sendet die angerufene MailBox nichts, schickt die Anruferin nach 10 Sekunden "`\r`" (nur "`\r`", kein "`\n`", da dieses auf einigen Betriebssystemen Probleme beim Einloggen verursachen kann[2]) und wartet wiederum auf die Einloganforderung.

Bei erkannter Login-Anforderung sendet die Anruferin den Benutzernamen "`zconnect\r`". Erhält sie darauf innerhalb von 10 Sekunden keine Antwort, sendet sie "`\r`" und wartet wieder auf einen der o.g. Strings. Es sollten mindestens drei Versuche gestartet werden, bevor bei Mißerfolg die Verbindung abgebrochen wird.

Die Empfängerin antwortet bei Erhalt des Antwortstrings durch Senden der Passwortabfrage, in der einer der Strings "`word`", "`WORD`", "`wort`" oder "`WORT`" enthalten sein muß. Erhält die Empfängerin keine Reaktion von der Senderin, so sendet diese von sich aus alle 2 Sekunden erneut den zuletzt gesendeten String.

Die Senderin schickt darauf das Passwort "`0zconnec\r`". Wird innerhalb von 10 Sekunden nicht geantwortet, sendet sie "`\r`" und wartet nun erneut auf die Passwortabfrage oder die Login-Frage.

Nach dem Erhalt des "`0zconnec\r`" startet die Empfängerin die ZCONNECT-Übertragung. Dazu sendet sie dreimal den String "`BEGIN\r`" mit jeweils 0,5 Sekunden Pause dazwischen und 1 Sekunde Pause danach. Dieses signalisiert der Anruferin, daß das Login gelungen ist und der Transfer beginnen kann. Die Verbindung wird abgebrochen, wenn das Login nicht innerhalb von 2 Minuten vollendet wird.

Nach Erhalt des Strings "`BEGIN\r`" geht auch die Anruferin in den Transfermodus.

---

[1] im Gegensatz z.B. zu UUCP

[2] z.B. OS-9

II.4 Grundlagen des ZCONNECT-Protokolls

Eins der Hauptprobleme eines geregelten Filetansfers zwischen Systemen mit wechselnden Anforderungen ist eine möglichst flexible Definition des Datenübermittlungsprotokolls. Beim alten ZERBERUS-Netcallprotokoll (das im Hitchhikers Guide to Zerberus Netcalls' von Patrick Schaaf dokumentiert wurde und vom ZERBERUSMailBox Programm bis Version 4.0 sowie von den diversen ZERBERUS-netcallkompatiblen Programmen verwendet wurde) war es nicht möglich, mehr als ein Mailfile auszutauschen, Files zu requesten oder ein Teil der Daten von der Gegenstelle auch mal einfach löschen zu lassen, ohne sie zu übertragen.

Leider kann man sich ein flexibleres Protokoll nur durch gestiegene Komplexität erkaufen. Wir haben zwar versucht, es so einfach wie möglich zu gestalten, jedoch wurde bei der Abwägung Flexibilität – Einfachheit, der Flexibilität der Vorzug gegeben.

Anforderungen:

- Flexibel, d.h. die Protokollinformationsdaten müssen erweitert werden können - ohne wiederum ein neues Protokoll definieren zu müssen
- Verschiedene Übertragungsprotokolle. Die Daten müssen mit verschiedenen Protokollen (z.B. ZMODEM, BIMODEM, `ftp`) übertragen werden können - ohne daß die Systembetreuung dies jeweils einstellen muß.
- Neue Anforderungen automatisch mit der Gegenseite absprechen. Sollte z.B. das Packerformat gewechselt werden, müssen beide automatisch auf das neu eingestellte Format gehen - oder sich auf dasselbe Format einigen (falls das Gewünschte bei der Gegenseite nicht verfügbar ist), ohne daß die Sysstembetreuung es einstellen muß und ohne daß deshalb ein Netcall mißglückt.
- Die Anruferin wählt aus, was sie haben will. Diejenige, die die Leitungskosten trägt, sagt auch, was sie wie haben will und was nicht, z.B. Eilmails abliefern, aber die öffentlichen Nachrichten bis zum nächsten Anruf liegenlassen (und in der Zwischenzeit vorpacken).
- Teilweise korrekt übertragene Daten müssen soweit wie möglich akzeptiert werden. Beim nächsten Anruf werden nur die fehlerhaften Daten wiederholt.
- Das Protokoll selber muß so ausgelegt sein, daß es auch auf Verbindungwegen funktioniert, die nicht vollkommen datentransparent sind. Bestes Beispiel ist das normale Datex-P Profil. Bedingung: es gelten nur CR und alle ASCII-Zeichen von hex 20 bis hex 7E.

Nach dem Login werden Informationen nach einem relativ einfachen, aber flexiblen Verfahren ausgetauscht. Das Protokoll hat ein gleichbleibendes Grundmuster:

Es werden *abwechselnd* Blöcke untereinander ausgetauscht bis ein anderes Protokoll (zum Dateitransfer) aufgerufen wird. Danach geht der gegenseitige Austausch von Blöcken weiter. Ein Block wird von der anderen Station positiv bestätigt (der Anwort-Block kann natürlich wiederum neue Informationen enthalten) oder abgelehnt. Falls keine Bestätigung empfangen wird, wird derselbe Block nach einer Timeoutzeit von 30 Sekunden noch einmal gesendet und wieder auf Bestätigung gewartet.

Die Blöcke sind durchlaufend numeriert, damit keine Verwechslungen auftreten können.

II.4.1 Aufbau eines Blocks

Das gesamte ZCONNECT-Online-Protokoll basiert auf dem Austausch von Blöcken. Es handelt sich dabei um Datenpakete variabler Länge - maximal ist ein Block (einschließlich der Blockende- Zeichen) 32 kByte groß. Jeder Block ist in mehrere Zeilen unterteilt, die durch ein "`\r`" (CR) getrennt werden. In jeder Zeile befindet sich ein Header (wie auch in den ZCONNECT-Daten, siehe unten), also eine alphanumerische Kennung

gefolgt von einem Doppelpunkt ":" sowie der Nutzinformation bis zum Zeilenende. (Im Gegensatz zum Nachrichtenformat sind hier keine Leerzeichen oder Tabulatoren zwischen dem ":" und dem Inhalt erlaubt.)

In einem Block sind nur folgende ASCII-Zeichen zugelassen:

- Hex. 20 (dezimal 32) bis Hex. 7E (dez 127) einschließlich
- CR, Hex. 0d (dezimal 13)

Alle anderen Zeichen werden als nicht übertragen gewertet, also von der Empfängerin ignoriert - so auch Linefeeds.

Das Zeichen CR gilt als Zeilentrenner. Zwei direkt aufeinander folgende CR's markieren das Block-Ende.

`\r` Zeilentrenner
`\r\r` Blockende

Es empfielt sich aus Sicherheitsgründen, auch vor dem Block-Anfang zwei Zeilentrenner zu senden, da einzelne Zeilentrenner vor einem Block ignoriert werden.

Header dürfen mehrfach in einem Block auftreten (also mehrere Zeilen mit derselben Kennung), wenn das sinnvoll ist.

Über jeden Block wird eine Prüfsumme auf Basis eines 16-Bit CRC nach X-MODEM Polynom errechnet. Diese Prüfsumme wird dann im Block selber mitgesendet. Dies geschieht auf einer Zeile mit der Kennung "`CRC`". Die Prüfsumme wird in großgeschriebenen Hexadezimalzahlen (vierstellig mit führenden Nullen) angegeben. Diese Zeile sowie alle "`\r`" gehen nicht mit in die Prüfsumme ein. Der CRC muß daher nicht an einer bestimmten Position im Block stehen. Der CRC muß auch bei gesicherten Verbindungen übertragen werden.

Weiterhin erhält jeder Block eine eindeutige Kennung, die seine Verwendung / Bedeutung im Protokollablauf definiert. Diese Zeile hat die Kennung "`STATUS`". Als Daten in dieser Zeile gibt es derzeit:

`BLK`*n* Datenblock (*n* von 1 bis 4, je nach Phase des Protokolls)

`ACK`*n* Empfangsbestätigungen auf den entsprechenden `BLK`*n* (daher auch *n* von 1 bis 4)

`NAK0` negative Empfangsbestätigung: der Block wurde empfangen, ist aber ungültig (z.B. falscher CRC durch Übertragungsfehler)

`TME`*n* Umschaltungsblock für Erhaltung der Übertragungsrichtung (*n* von 1 bis 4)

`EOT`*n* Protokollübergänge und Abschlußzeichen (*n* von 1 bis 4)

`BEG`*n* Bestätigungen, daß das Datenpacken beendet wurde (*n* von 5 bis 6)

`EOT`*n* Empfangsbestätigung auf empfangene `BEG`s, Protokollübergänge und Abschlußzeichen (n von 5 bis 6)

Hier ein Beispiel für einen gültigen Block (Das "`\r`" am Zeilenende steht für das ¡CR¿ als Zeilentrennung):

```
SYS:BI-LINK\r
SYSOP:Postmaster\r
TEL:1 +49-521-19304\r
PROTO:ZMODEM\r
PASSWD:GUEST\r
STATUS:BLK1\r
CRC:F32C\r
\r
```

Hinweis: Der angegebene CRC-Wert ist falsch, der Block ist als `BLK1` auch nicht komplett, er dient hier nur als Beispiel für den Block-Aufbau.

II.4.2 Protokollablauf

Die Senderin fordert Information durch einen BLK1 an. Die Empfängerin bestätigt den korrekten Empfang durch Senden eines ACK1; eine falsche Prüfsumme wird durch Senden eines NAK0 angezeigt. In einem solchen Fall muß der BLK1 wiederholt gesendet werden.

Ein richtig empfangener ACK1 auf der Senderseite wird durch Senden eines TME1 bestätigt, um die Gegenseitigkeit des Protokolls beizubehalten. Ein TME wird nicht durch ein ACK bestätigt, sondern durch Senden eines BLK (BLK2 in diesem Fall). Ab jetzt läuft alles wie vorher, nur mit vertauschten Rollen.

Es passiert also abwechselnd ein Austausch von BLK, ACK, TME, BLK, ACK usw. bis zu ACK4, danach kann mit einem TME4 von der Empfängerin die Fortsetzung des Protokolls mit einem BLK1 gefordert werden[3]. Durch Senden von drei EOT4 hintereinander (mit einer Sendepause von mindestens 1 Sekunde dazwischen) wird der Beginn eines anderen Übertragungsprotokolls (z.B. ZMODEM um eine Datei zu übertragen) oder des "Warten auf Aktion" markiert.

Danach gehen beide wieder auf das ZCONNECT-Protokoll. Wenn die Übertragung fehlerhaft gewesen sein sollte, erfolgt hier ein sofortiger Verbindungsabbruch und bei Bedarf eine erneute Anwahl des Systems. Ein Verbindungsabbruch erfolgt auch, wenn anhand des (optionalen) BYTE Headers festgestellt wird, daß das zu übertragende Datenpaket nicht vollständig empfangen wurde. War die Übertragung erfolgreich, so sendet die Angerufene solange NAK0, bis sich der Sender wieder mit einem BLK1 meldet. Die Daten gelten als erfolgreich übertragen, wenn dieser BLK1 nun vom Empfänger mit einem ACK1 bestätigt wurde. Ohne diesen Handshake nach dem Protokolltransfer wird jeweils das zuletzt ausgetauschte Datenpaket (beim Vollduplex-Protokoll empfangenes und gesendetes Datenpaket) beim nächsten Anruf erneut übertragen, auch wenn das Übertragungsprotokoll keinen Fehler gemeldet hat, da die Übertragungsprotokolle nicht unbedingt auf beiden Seiten einen Fehler melden müssen. Die in einem früheren Zyklus übertragenen Mailpakete werden nicht erneut übertragen.

Zu beachten ist, daß EOT4 bzw. EOT6 nur ein Endzeichen für die Synchronisation der Wartezeit ist. Die Information, ob und welches Übertragungsprotokoll gestartet werden soll, wurde schon vorher in BLK4 und / oder BLK3 übertragen.

Das Warten auf Aktionen vor der Fileübertragung ist optional und erfolgt nur, wenn entweder Senderin oder Empfängerin mit dem Kommando WAIT eine Wartezeit gefordert haben. Ein Wartezeit wird meist nur dann gefordert, wenn die Bereitstellungszeit länger als 10 Sekunden dauert. Haben sowohl Senderin als auch Empfängerin WAIT: 0 vereinbart (oder gar kein WAIT Kommando geschickt), so beginnt der Protokolltransfer nach dem dritten EOT4.

Wenn z.B. die Senderin warten soll, damit die Empfängerin Zeit hat, ihr Mailarchiv

[3] Durch Senden eines TME4 kann daher auch der Transfer eines Mailpaketes abgelehnt werden - dieses wird bei ZERBERUS Version 5.2 derzeit verwendet, wenn keine Daten zur Übertragung bereitliegen. Neuere ZCONNECT-Implementationen kündigen dieses korrekter durch einen leeren PUT Header an.

ZCONNECT beim Austausch der Steuerinformationen:

```
Send.: BLK1    TME1    ACK2    BLK3    TME3    ACK4    BLK1
                                                          u.s.w.
Empf.:     ACK1    BLK2    TME2    ACK3    BLK4    TME3    ACK1
```

ZCONNECT mit Packen und externem Übertragungsprotokoll:

```
Send.:     ACK4             -Packen-    EOT5-EOT5-EOT5 BEG6             ZMODEM
Empf.: BLK4   EOT4-EOT4-EOT4 -Packen- BEG5               EOT6-EOT6-EOT6 ZMODEM
```

Abbildung II.1: Darstellung der Reihenfolge der übertragenen ZCONNECT-Blöcke

zu packen (nur die seit dem letzten Packen hinzugekommenen Daten im ZCONNECT werden vorgepackt gespeichert und neue adaptiv gepackt), stellt die Senderin nach dem Senden des letzten EOT4 die Übertragung ein und komprimiert ihr Archiv. Wenn sie fertig ist, zeigt sie der Senderin dies durch Senden eines BEG5 (vorher Empfangspuffer löschen) an. Diese zeigt durch dreimaliges Senden eines EOT5 an, daß sie verstanden hat.

Hatte auch die Senderin etwas zu packen, so sendet diese nach dem Packen zunächst ein BEG6, welches von der Empfängerin mit drei EOT6 bestätigt wird, sofern diese zum Empfang schon bereit war. Ist diese dagegen noch beim Bereitstellen, wird sie dagegen zunächst mit einem BEG5 melden, welches von der Empfängerin dann mit EOT5 bestätigt wird, hiernach erfolgt dann nochmal der BEG5/EOT6-Austausch. Nach dem Übertragen des EOT6 beginnt dann der eigentliche Datentransfer. Hatte die *Senderin* nichts zu packen, so entfällt der BEG5/EOT5 Austausch.

Nun können beide mit dem eigentlichen Übertragen (externes Fileprotokoll) beginnen. Wenn ein beidseitiger Datenaustausch vereinbart wurde (PUT: und GET: im gleichen Block) und *kein* bidirektionales Protokoll verwendet wird, sendet hier zunächst die Anruferin, direkt danach sendet die angerufene MailBox.[4] Die Anruferin hat darauf zu achten, daß generell immer nur ein Datenpaket angefordert wird - die gleichzeitige Anforderung eines Mailpaketes und eines Filerequests ist unzulässig. Ein bidirektionaler Datentransfer kann von der Anruferin prinzipiell mit EXECUTE: N abgelehnt werden. In diesem Falle muß die Anruferin die Kommandos aufsplitten und der Angerufenen einzeln vorlegen.

II.5 Header für Systeminformationen

Nachfolgend werden die Kennungen erläutert, die in der ersten Phase (BLK1 bis ACK4) übermittelt werden, um Daten über die beiden Systeme auszutauschen.

Bei einigen Headern *muß* in der Datenzeile eine Portnummer angegeben werden. Dazu wird die Portnummer, gefolgt von einem Leerzeichen, am Anfang der Nutzinformation angegeben. Sind die Informationen auf alle Ports anwendbar, wird 0 als Portnummer angegeben.

ARC +Portnummer, Pflicht

Mögliche Datenkompressionen. Definiert sind bisher:

Kürzel	Name	Dateiendung
ARC	Arc/Pak	.arc
ARJ	Arj	.arj
LHARC	Lh1/2	.lzh
LHA	Lh1-5	.lha
RAR	Rar	.rar
ZOO	Zoo	.zoo
ZIP	Zip	.zip
ZIP2	Zip V2.0	.zi2
COMPRESS	Compress	.Z
GZIP	GNU-Zip	.gz
TAR-COMPRESS	Tar und Compress	.tar.Z, .tz
TAR-GZIP	Tar und GNU-Zip	.tar.gz, .tgz
RM	Remove	entfällt
NONE	Keine Kompr.	.non

Auch hier werden Alternativen durch Leerzeichen getrennt hintereinandergestellt. Die Reihenfolge ist dabei relevant: die bevorzugten Packer stehen vorne.

[4] Die Simulation eines Vollduplex-Protokolles ist nur aus Kompatibilitätsgründen zu den ersten Zconnect-Versionen definiert. Neue Zconnect-Implementation sollten einen Vollduplex-Transfer nur dann anfordern, wenn auch ein Vollduplex-Protokoll in der Informationsphase vereinbart wurde

Üblicherweise werden Packer auf allen Ports gleichartig verfügbar sein, sodaß nur ein ARC-Header mit Portnummer 0 gesendet wird.[5] In Ausnahmefällen (verschiedene Hard / Software auf verschiedenen Ports) kann dies aber auch anders sein.

ARCEROUT optional

Der Packer, mit dem die Daten zur Zeit gepackt vorliegen. Kodierung wie bei ARC. Die Angerufene sollte in BLK2 mit ihren Systemdaten der Anruferin unbedingt mitteilen, mit welchem Packer die Daten vorkomprimiert wurden. Anhand der Systemdaten entscheidet die Anruferin dann, welche Packer endgültig verwendet werden und überträgt diese endgültigen Parameter im BLK3. Unter ARCERIN steht hierbei dann der Packer, den die Angerufene verwenden muß (bei Bedarf muß dann in der Vorbereitungsphase umgepackt werden) und bei ARCEROUT der Packer, den die Anruferin selber verwendet.

ARCERIN Optional

Der Packer, mit dem die Daten bei der Übertragung gepackt sein sollen. Dieser wird nur dann von ARCEROUT abweichen, falls ARCEROUT auf dem empfangenden System nicht extrahiert werden kann. In diesem Fall fordert das betroffene System mit ARCERIN die Gegenstelle zum Umpacken auf. Der Wert für ARCERIN wird aus dem ARC-Header der Gegenstelle ermittelt, so daß dieser Packer auf jeden Fall für beide verfügbar ist. Sowohl ARCEROUT als auch ARCERIN bezeichnen die Packer immer aus Sicht der Senderin des Headers - ARCERIN des Anrufers entspricht ARCEROUT der Angerufenen.

CRYPT +Portnummer, optional

Hiermit sind alle Verschlüsselungsprogramme, die für die Mailkodierung verwendet werden können, gemeint.

Kürzel	Name
DES	DES 56Bit DoD-Standard (lowtech)
PGP	RSA Public-Key-Verschlüsselung (hightech)

Sind mehrere Crypt-Möglichkeiten vorhanden, gilt sinngemäß das unter ARC Gesagte. Eine nicht existente CRYPT Zeile besagt, daß die Daten nicht verschlüsselt sind, die CRYPT Zeile gilt generell sowohl für die empfangenen als auch die gesendeten Daten. Die Passwort-Vereinbarung erfolgt offline; Eine Public-Key-Passwortaustausch wird noch definiert. Ein evt. notwendiges Entcrypten / Umcrypten von Daten sollte generell offline erfolgen.

DOMAIN optional

Internet-Domains des Systems. Falls mehrere gültige Domains vorhanden sind, werden diese durch Leerzeichen oder Semikolon getrennt aufgeführt.

Beispiel:

```
DOMAIN:zer.de comlink.de zer.sub.org\r
```

ISO2 +Portnummer, Pflicht

Mögliche Verbindungsarten:

[5] ZERBERUS Vers 5.2 sendet derzeit prinzipiell die Informationen für alle Ports separat und interpretiert Portnummer 0 nicht

Kürzel	Verbindungsart
V.21	300 Bps CCITT
V.22	1200 Bps CCITT
V.22bis	2400 Bps CCITT
V.32	9600 Bps CCITT
V.32bis	14400 Bps CCITT
V.34	28800 Bps CCITT
V.34T	28800 Bps Terbo
V.FC	28800 Bps FastClass
PEP	15000 Bps PEP-Modus
HST	14400 Bps HST-Modus
V110	ISDN-Protokoll
BUNDLE	ISDN-Protokoll
X.75SLP	ISDN-B-Kanal Protokoll
HDLC	ISDN-B-Kanal Protokoll
BITTRANSP	ISDN-B-Kanal Protokoll
SNA-SDLC	ISDN-B-Kanal Protokoll
X.75BTX	ISDN-B-Kanal Protokoll
X.25	Datex-P
Z.16	16800 Bps Zyxel
Z.19	19200 Bps Zyxel

Sind mehrere Verbindungsarten möglich, werden diese durch je ein Leerzeichen getrennt hintereinandergestellt. Für jeden TEL-Header muß ein zugehöriger ISO2-Header gesendet werden, es sei denn, alle Ports haben gleiche Fähigkeiten: dann wird ein ISO2-Header mit Portnummer 0 gesendet.

ISO3 +Portnummer, optional, default: Transparent

Angabe der Fehlerkorrekturmöglichkeit und Möglichkeit für Datenkorrektur auf dem Verbindungsweg.

Kürzel	Verbindungsprotokoll
MNP	bis Level 4 (nur Fehlerkorr.)
MNP5	Level 5 bis 9 (Mit Datenkompr.)
MNP10	
V.42	Fehlerkorrektur
V.42bis	Datenkompression
T70NL	ISDN-B-Kanal Protokoll
ISO8208	ISDN-B-Kanal Protokoll
T90	ISDN-B-Kanal Protokoll
TRANSPARENT	Kein Ebene 3 Protokoll

Falls mehrere Alternativen bestehen, werden diese durch je ein Leerzeichen getrennt hintereinandergestellt. Wurden mehrere TEL-Header gesendet sind auch entsprechende ISO3-Header zu senden (vergleiche auch ISO2).

KOORDINATEN optional

Standort in geographischer Länge und Breite. Format wie in folgendem Beispiel:

```
KOORDINATEN:53 11 N / 10 44 E (city)\r
```

Die Ergänzung (city) in diesem Beispiel bezeichnet hierbei, daß die Koordinaten die einheitlichen, für die jeweilige Stadt gültigen Koordinaten sind, die sie zum Beispiel in jedem handelsüblichen Atlas (im Register den Stadt-Namen suchen) oder bei Ihrer Stadtverwaltung erfahren.

MAILER +Portnummer, optional, Default: nur ZCONNECT

Hier sind alle Formate zum Verbindungsaufbau definiert, die das System versteht. Folgende Einträge sind zur Zeit bekannt:

Kürzel	Verbindungsprotokoll
ZCONNECT3.0	Das ZCONNECT-Datenformat vom 3. Aug. 92
ZCONNECT3.1	Das im folgenden Kapitel beschriebene Datenformat
ZCONNECT	steht für ZCONNECT3.0 und ZCONNECT3.1
ZNETZ	das alte ZERBERUS-Protokoll
FTS0001	FidoNet
FSC0056	EMSI-Standard im Fido-Net
MausTausch	Die Schnittstelle zum MausNet

Mehrere Möglichkeiten werden wie unter ARC erläutert dargestellt.

MAILFORMAT +Portnummer, optional, Default: nur ZCONNECT

Hier sind alle Mailformate aufgeführt, die das System versteht. Folgende Einträge sind zur Zeit bekannt:

Kürzel	Datenformat
ZCONNECT3.0	Das ZCONNECT-Datenformat vom 3. Aug. 92
ZCONNECT3.1	Das im folgenden Kapitel beschriebene Datenformat
ZCONNECT	steht für ZCONNECT3.0 und ZCONNECT3.1
ZNETZ	das alte ZERBERUS-Format
RFC1036	oft nur “UUCP” genannt, das News-Format wie es z.B. in UUCP-Systemen verwendet wird.
X400	ISO / OSI Standard

Mehrere Möglichkeiten werden wie unter ARC erläutert dargestellt.

MAPS optional

Namen weiterer Userinnen, die Maps benutzen dürfen - zusätzlich zur Systembetreuung, die bereits im SYSOP Header benannt wird. Hier werden die Benutzerinnennamen von Cosysops aufgeführt (einem pro Header), die für dieses System auch Maps-Bestellungen abgeben dürfen.[6]

[6] ZERBERUS Version 5.2 schneidet diese Zeile nach dem 40. Zeichen ab - ZERBERUS Version 5.3 akzeptiert hier 60 Zeichen

PASSWD Pflicht

Passwort, das benötigt wird, um die Verbindung aufrechtzuerhalten. Wird zwischen den beiden Systemen zum ersten Mal eine Verbindung aufgebaut und wurde noch kein Passwort ausgetauscht, so generiert der Sender ein Passwort, welches von der Empfängerin übernommen und ab dem nächsten Verbindungsaufbau verwendet wird. Das Passwort wird allerdings erst nach dem erfolgreichen Abschluß der Informationsphase gespeichert. Ist dagegen bei der Angerufenen ein Passwort definiert, so muß dieses stimmen (Groß/Kleinschrift wird beachtet). Sendet die Anrufende als Passwort "GUEST" und ist sie der Angerufenen nocht nicht bekannt, so handelt es sich um einen einmaligen Datenaustausch und die Anruferin wird nicht in die Systemliste der Angerufenen aufgenommen.

Das Passwort ist maximal 10 Zeichen lang.

PORT Pflicht

Portnummer, auf der sich die aktuelle Anruferin/Angerufene befindet. Falls ein System nur einen Port hat, gibt es "1" als Portnummer an.

POST optional

Vollständige Postadresse des Systems, falls jemand einen Brief per Post schicken möchte.

PROTO +Portnummer, Pflicht

Alle Übertragungsprotokolle, die für die Übertragung von Daten benutzt werden können (nicht alle können bei jedem Modem und/oder Anschluß genutzt werden, z.B. BiModem nicht im PEP-Modus).

Kürzel	Übertragungsprotokoll
XMODEM	Xmodem
YMODEM	Ymodem
ZMODEM	Zmodem
SEALINK	Sealink
KERMIT-B	Kermit im Binärmodus
BIMODEM	BiModem (bidirektional)
HSLINK	HighSpeedLink (bidirektional)
ACOPY	Acopy (ISDN Protokoll)
HYDRA	Bidirektionales Protokoll
EFT	ISDN-File-Transfer-Standard
ZMODEM8K	8K-Variante des ZModem.
NCOPY	Network Copy

Mehrere Möglichkeiten werden durch je ein Leerzeichen oder Semikolon getrennt hintereinandergestellt. Hier gilt im übrigen sinngemäß das unter ARC Gesagte (s.o.).

SERNR Default: "0", optional

Seriennummer der verwendeten Software. Jedes System kann bei der Auslieferung eine ID erhalten, die kein anderes System hat. Sollten zwei gleiche Seriennummern miteinander versuchen zu kommunzieren, wird der Versuch abgebrochen. Sendet eines der Systeme diesen Header nicht, wird weiter kommuniziert.

SYS Pflicht

Sowohl Anruferin als auch angerufenes System müssen diesen Header mit dem Systemnamen genau einmal senden, ansonsten wird die Verbindung abgebrochen. Die Anruferin prüft, ob das System, das sie mit dieser Verbindung erreichen wollte, sich mit eben diesem Namen meldet. Geschieht das nicht, wird ebenfalls abgebrochen.

SYSOP Pflicht

Name der Userin in der Box, an die Fragen zum System u.s.w. geschrieben werden können, z.B. “Postmaster”, “Zentrale” oder “Sysop”.

TEL +Portnummer, Pflicht

Nummer des Telefonanschlusses. Die Telefonnummer wird nach internationaler Notation angegeben, also zuerst ein Pluszeichen, dann der Ländercode (z.B. 49 für BRD), ein Bindestrich, die Vorwahl ohne führende Null, noch ein Bindestrich und dann die Anschlußnummer. Beispiel:

```
TEL:1 +49-521-19304\r
TEL:2 +49-521-19300\r
```

Für Datex-P oder Telex werden die Nummern nach den dort allgemein gültigen Formen angegeben.

TELEFON optional

Die Voice-Telefonnumer, Format wie unter TEL: beschrieben, jedoch ohne die hier unsinnige Portnummer. Falls unter der angegebenen Telefonnummer ein Anrufbeantworter vorhanden ist, wird der Nummer ein Q nachgestellt. Mehrere Nummern werden durch ein Semikolon oder durch Leerzeichen getrennt.

LOGOFF optional

Hiermit kann bei Unverträglichkeiten oder falschem Passwort eine Verbindungstrennung angekündigt werden, die erfolgen wird, nachdem die Header BLK4 / ACK4 / TME4 ausgetauscht wurden. Der Logoff kann von beiden Systemen jederzeit angefordert werden. Als Parameter wird der Grund der Verbindungstrennung als Klartext-Fehlermeldung übergeben. Beispiele: “Falsches Passwort” oder “Seriennummern identisch”. Der Logoff kann von beiden Seiten jederzeit angefordert werden.

Im folgenden nun ein Beispiel für einen korrekten Verbindungsaufbau mit dem Austausch aller notwendigen Blöcke aus der Sicht der Anruferin. Die CRC-Checksummen der Datenblöcke stimmt dabei nicht immer, da teilweise eine Formatierung der Daten notwendig war.

Während des Datenaustausch kann diejenige, die auf einem Block wartet, die Gegenseite mit einem NAK0-Header zu einer Wiederholung des jeweils zuletzt gesendeten Blockes (BLK1-BLK4) auffordern. Bei einer verstümmelten Bestätigung eines Blocks (ACK1-ACK4) wird der zuletzt gesendete Block nach spätestens 30 Sekunden erneut wiederholt. Nach 15 Wiederholungen erfolgt ein Verbindungsabbruch.

Empfangen:

```
BEGIN
BEGIN
```

Gesendet:

```
Sys:SYSTEM1
Sysop:SYSOP
SerNr:Z001
Post:Wolfgang Mexner, Strasse, Stadt
Port:1
Tel:1 +49-5509-2556
Domain:zer.sub.org;zer
Maps:
ISO2:1 V.32bis V.32 V.22bis V.22 V.21
ISO3:1 V.42bis V.42 MNP5 MNP
Arc:1 ZIP2 ZIP ARJ ARC ZOO LHA NONE
Proto:1 HSLINK ZMODEM XMODEM BIMODEM
```

```
Passwd:JMFP5Y5GZE
Telefon:+49-5509-2556Q +49-5509-919010
Status:BLK1
ArcerOut:ZIP2
Mailer:1 ZCONNECT FIDO JANUS UUCP ZNETZ
CRC:A304
```

Achtung: Der angegebene CRC stimmt nicht, da der Header von Hand bearbeitet wurde.

Empfangen:

```
Status:ACK1
CRC:EA3C
```

Gesendet:

```
Status:TME1
CRC:F974
```

An dieser Stelle trat ein Timeout bei der Angerufenen auf, der letzte Block wird daher wiederholt.

Empfangen:

```
Status:ACK1
CRC:EA3C
```

Gesendet:

```
Status:TME1
CRC:F974
```

Empfangen:

```
Sys:SYSTEM2
Sysop:SYSOP
SerNr:Z000
Post:ZERBERUS GmbH, Marktstr. 18, 33602 Bielefeld
Port:1
Tel:1 +49-521-9680869
Tel:2 +49-521-68000
Domain:zer.sub.org;zer;comlink.de
Maps:
ISO2:1 ISDN
ISO2:2 V.32bis V.32 V.22bis V.22 V.21 ISDN
ISO3:1 V.42bis V.42 MNP5 MNP
ISO3:2 V.42bis V.42 MNP5 MNP
Arc:1 ZIP2 ZIP ARJ ARC ZOO LHA NONE
Arc:2 ZIP2 ZIP ARJ ARC ZOO LHA NONE
Proto:1 HSLINK ZSXW ZMODEM
Proto:2 HSLINK ZMODEM XMODEM BIMODEM
Telefon:0521/65566 Fax 61172
Status:BLK2
ArcerOut:ZIP
MAILER:1 ZCONNECT FIDO JANUS UUCP ZNETZ
MAILER:2 ZCONNECT FIDO JANUS UUCP ZNETZ
CRC:9CF6
```

Achtung: Der angegebene CRC stimmt nicht, da der Header von Hand bearbeitet wurde.

Gesendet:

```
Status:ACK2
CRC:EA3F
```

Empfangen:

```
Status:TME2
```

```
CRC:F977
```

Gesendet:

```
Proto:HSLINK
Status:BLK3
ArcerIn:ZIP
ArcerOut:ZIP2
CRC:8036
```

Empfangen:

```
Block empfangen:
Status:ACK3
CRC:EA3E
```

Gesendet:

```
Status:TME3
CRC:F976
```

Empfangen:

```
Status:BLK4
CRC:4E85
```

Gesendet:

```
Status:ACK4
CRC:EA39
```

Empfangen:

```
Status:TME4
CRC:F971
```

II.6 Header zur Verbindungssteuerung in der Datenaustausch-Phase

Nachdem die Systeminformationen ausgetauscht und bestätigt wurden, beginnt der Datenaustausch (sofern nicht ein Logoff angefordert wurde). In der nun folgenden Phase werden die zuvor vereinbarten Packer und das Übertragungsprotokoll nicht mehr verändert. Die Steuerung des Datenaustausches erfolgt in dieser Phase immer durch das anrufende System. Die Anruferin hat nur die Möglichkeit, einzelne Kommandos abzulehnen oder nur teilweise auszuführen. Bei einem Vollduplex-Protokoll werden immer gleichzeitig ein Empfangs- und ein Sendekommando abgearbeitet; bei einem Halbduplex-Protokoll werden sequentiell zunächst alle Empfangs- und dann alle Sende-Kommandos abgearbeitet. Mehrere Empfangs- bzw. Sende- Kommandos in einem `BLK1` sind unzulässig, da die Daten nicht im Batch-Transfer übertragen werden.

Wird neben den bekannten Kommandos eine unbekannte Aktion gefordert, so wird diese vom angerufenen System ignoriert - evtl. erfolgt noch eine `EXECUTE: N` Meldung. Das ein Befehl nicht erkannt bzw. ausgeführt wurde, kann daran erkannt werden, daß anstelle des `EOT4`-Headers am Ende der Sequenz `BLK1`-`BLK4` einfach ein `TME4` gesendet wird.

Bisher sind für die Anruferin folgende Kommandos für `BLK1` definiert:

`GET` optional

Die anrufende MailBox möchte Daten von der angerufenen abholen.

Buchstabe	angeforderte Mailpakete
P	Persönliche Mails und Mails mit sowohl Brett- als auch privaten Empfängern abholen
E	nur Eilmails abholen.
B	Brettnachrichten sollen abgeholt werden.
F	Fehlermails abholen

Die Buchstaben können auch kombiniert werden. Mit "GET:PEBF" werden beispielsweise alle Mails angefordert. Auf das Kommando GET kann das angerufene System im Antwort-Block mit der Meldung PUT:PBEF der Anruferin mitteilen, welche Mailtypen zur Abholung bereitliegen. Eine leere PUT- Antwort bedeutet, daß keine Daten von den mit GET gewünschten Typen verfügbar sind. Diese Option ist besonders wichtig, wenn die angerufene MailBox die Daten in Blöcken einer einstellbaren Größe archiviert. So wird bei dem Kommando GET:PBEF nur ein Bruchteil der gespeicherten Daten übertragen. Die Anruferin wird daher das Kommando GET:PBEF solange wiederholen, bis ein leerer PUT Header als Antwort kommt.[7]

PUT optional

Format wie GET. Die anrufenden MailBox möchte Daten der Typen PBEF zur angerufenen übertragen. Möchte eine anrufende MailBox also alle verfügbaren Daten von einem System abholen, so wird sie zunächst das Kommando GET:PBEF und dann das Kommando PUT:PBEF erteilen.

DELETE optional

Das einzige zusätzliche Kommando, was jederzeit parallel zu einem Empfangs- und Sende-Kommando ausgeführt werden kann, ist das DELETE-Kommando. Dieses Kommando löscht alle Mails mit der Extension .PRV, .KOM, .BRT, .ERR oder .EIL

Buchstabe	bewirkt Löschung von
P	Persönliche Mails und Mails mit sowohl Brett- als auch privaten Empfängern
B	Brettnachrichten (öffentliche Mails)
E	Eilnachrichten
F	Zurückgeschickte fehlerhafte Nachrichten

Dies wird vom anderen System ignoriert, falls es sich um Mails handelt, die über dieses System an ein anderes geroutet werden sollen; Routmails (PMs) dürfen niemals gelöscht werden. Dieses Kommando wird daher meist nur für Points verfügbar sein. Das DELETE-Kommando hat Vorrang vor einem GET-Kommando.

FORMAT optional, Default: 'ZCONNECT'

Dieser Header tritt nur zusammen mit einem PUT Header auf. Die Kennung gibt an, in welcher Form die Mails / News vorliegen.

Dieser Header ist nur gültig in der Datenphase der Online-Kommunikation. Es dürfen nur Formate angegeben werden, die von der Gegenseite im MAILER Header aufgeführt wurden. Die Gegenseite kann (selbstverständlich) die Übertragung ablehnen - es gibt aber keine Möglichkeit, der anderen Seite ein anderes

[7] Ältere Systeme schicken keinen PUT Header als Antwort auf einem GET-Header. Hier muß solange das GET-Kommando geschickt werden, bis Anstelle des EOT4-Headers zum Einleiten des Protokolltransfers ein TME4-Header geschickt wird

Datenformat vorzuschreiben oder eine Umkodierung anzufordern. Zwei ZCONNECT Systeme werden niemals vollautomatisch ein anderes Datenformat als ZCONNECT zur Kommunikation untereinander wählen - dazu ist immer ein manueller Eingriff nötig.

Die Kennung ist dabei kodiert wie bei MAILER. Hiermit können sich z.B. zwei UNIX-Systeme RFC-1036 Datenpakete oder Fido-Systeme FTS-Pakete übermitteln.

FILEREQ optional

Datei von Empfängerin laden. Hier werden UNIX-Pfadnamen[8] verwendet. Der Dateiname "/INFO/INHALT" ist reserviert für die Gesamtübersicht aller verfügbaren Dateien. Es gibt *keine* Regeln für Dateinamen in diesem Zusammenhang, die Empfängerin muß auf alles vorbereitet sein (z.B. Leerzeichen und Sonderzeichen) und entsprechend ihren lokalen Konventionen wandeln.

Die Namen, die mit /INFO beginnen, sind dabei für bestimmte Systeminformationen reserviert, die – falls sie vorhanden sind – den hier stehenden Inhalt haben:

Kennung	Erläuterung
/INFO/LOGIN	Begrüssungstext beim Login
/INFO/MOTD	Bulletin (Message of the Day)
/INFO/Q-USAGE	Kurzanleitung der Box (Quick Usage)
/INFO/V-USAGE	Großanleitung der Box (Verbose Usage)
/INFO/HINTS	Bedienungshinweise (Hints & Twinkles)
/INFO/INTRO	Kurzanleitung für Gäste und NeuUser
/INFO/NETWORKS	Verfügbare Netze und deren "Ziele"
/INFO/SYSTEM	Selbstdarstellung der Box
/INFO/TIMES	Login Zeiten
/INFO/COSTS	Benutzergebühren etc
/INFO/KNIGGE	Nettikette(n)
/INFO/INHALT	Das Inhaltsverzeichnis des Fileservers.

Aus Kompatibilitätsgründen sollte auch mit /INHALT das Inhaltsverzeichnis des Fileservers geliefert werden.

FILESEND optional

File bei der Empfängerin speichern. Die Empfängerin entfernt evtl. den Directory-Namen aus dem angegebenen Namen oder verbietet die Übertragung. Für Dateinamen gilt das unter FILEREQ Gesagte.

PGP-KEYREQ optional

Mit diesem Header wird der ZCONNECT PGP-Key-Request realisiert. Hier wird die Adresse des Users angegeben, dessen Public-Key bei der nächsten Übertragung gesendet werden soll.

Falls vorhanden wird der Key der angegebenen UserIn nach dem folgenden BLK4 als Datei übertragen. Falls nicht wird die Ausführung abgelehnt (EXECUTE:N).

Im Antwort-Block BLK2 reagiert das angerufene System auf das in BLK1 übertragene Kommando. Es hat hierbei folgende Möglichkeiten:

[8] also / statt \, genau wie bei ZERBERUS-Brettern

EXECUTE Pflicht

Mit diesem Header melden beide Systeme, ob sie zu den angeforderten Aktion bereit sind. Die Anruferin kann hierbei die Ausführung seines eigenen Kommandos ablehnen, wenn ihr z.B. die Bereitstellungszeit zu groß ist.

Dieser Header kann daher sowohl von der angerufenen als auch der angerufenen MailBox gesendet werden. Er kann in BLK2-BLK4 als Antwort gesendet werden. Wird er mehrfach gesendet, so hat prinzipiell das Nein den Vorrang Soll ein Befehl von der angerufenen MailBox tatsächlich ausgeführt werden oder nicht.

Buchstabe	Bedeutung
N	Nein
J	Ja
L	Later. Soll später (nach dem Logoff) ausgeführt werden.

Wird EXECUTE:N bei einem kombinierten Empfangs / Sende-Kommando im Vollduplex-Betrieb vom angerufenen System gesendet, so ist nicht klar, ob beide Teilkommandos abgelehnt wurden. Daher müssen diese dann dem angerufenen System nochmals einzeln vorgelegt werden. Das EXECUTE:L Kommando sendet nur die Anruferin nach einer WAIT Meldung der Angerufenen, wenn die Online-Bereitstellung der Daten zu lange dauern würde. Die Daten werden in diesem Falle für den nächsten Anruf fertig bereitgelegt.

WAIT optional

Zeit in Sekunden, die wahrscheinlich für das Bereitstellen der Daten (z.B. wegen des Packens) gebraucht wird. Wenn dort ein "N" steht können die Daten nicht bereitgestellt (bzw. jetzt übertragen) werden. Bei einer WAIT:N Meldung wird ein Kommando prinzipiell abgelehnt - wenn z.B. ein bestimmtes Filerequest nicht möglich ist. Bei der Meldung WAIT:N ist gleichzeitig ein EXECUTE:N verknüpft.

BYTES optional

Größe des zu übertragenen Files in Bytes. Diese Headerzeile kann von beiden Seiten gesendet werden. Hiermit kann der Empfang eines zu großen Files abgelehnt werden, bevor die Festplatte überläuft.

FILE-CRC optional

Der Header FILE-CRC ist optional und kann in jeder BLK3 / BLK4-Phase nach Abschluß des Systemdatenaustauschs genau einmal gesendet werden. Falls er gesendet wird, gibt er (in Hex) den 32-bit CRC (nach CCITT / Z-MODEM Polynom) der nach BLK4 zu übertragenden Datei an. Falls dieser Header gesendet wurde, sollte die Gegenseite den CRC der empfangenen Datei prüfen und bei Unstimmigkeiten LOGOFF oder RETRANSMIT anfordern (z.B. 2 mal RETRANSMIT, dann LOGOFF)

LOGOFF optional

Hiermit können beide Seiten jederzeit das Ende der Verbindung anfordern. Im Normalfall wird der Logoff nach dem Transfer des letzten Datenblockes von der Anruferin mit der Meldung "Alle Daten uebertragen" angefordert. Genauso wie in der Info-Phase wird der Logoff-Text im Klartext übertragen.

RETRANSMIT optional

Der Header RETRANSMIT ist optional und kann in dem BLK1 oder BLK2 Paket genau einmal gesendet werden, daß direkt auf eine Dateiübertragung folgt (mit anderen Worten: dessen gültiger ACK / TME für die Gegenseite die Bestätigung der erfolgreichen Datenübertragung wäre). Er erklärt die letzte Übertragung als gescheitert und fordert eine erneute Übertragung an. Die Gegenseite darf diese

Datei daraufhin nicht löschen. Sie muß sie aber nicht unbedingt sofort noch einmal übertragen - dies kann nach Abschluß aller schon (lokal) geplanten Dateiübertragungen geschehen oder sogar erst im nächsten Netcall. `RETRANSMIT` gibt wie `LOGOFF` den Grund für die nochmalige Übertragung im Klartext an, z.B.

```
RETRANSMIT:CRC failed
```

`RETRANSMIT` kann auch aus beliebigen anderen Gründen angefordert werden. Allerdings ist dabei Vorsicht geboten. Beispiel für einen klaren Mißbrauch:

```
RETRANSMIT:out of disk space
```

In diesem Fall wäre ein schlichter Abbruch der Verbindung zu empfehlen - die Datei gilt auch dann als nicht korrekt übertragen und würde in einem der nächsten Netcalls wieder bereitstehen.

Im folgenden nun ein Beispiel für einen kompletten Headeraustausch in der Daten-Transferphase aus Sicht der Anruferin bei einem Vollduplex-Datentransfer. Die CRC-Summen sind hierbei wiederum der Form halber mit angegeben, stimmen aber nicht immer:

Gesendet:

```
Get:PEBF
Put:PEBF
Status:BLK1
CRC:9F7C
```

Empfangen:

```
Status:ACK1
CRC:EA3C
```

Gesendet:

```
Status:TME1
CRC:F974
```

Empfangen:

```
Status:BLK2
Wait:120
Put:PB
CRC:6C61
```

Gesendet:

```
Status:ACK2
CRC:EA3F
```

Empfangen:

```
Status:TME2
CRC:F977
```

Gesendet:

```
Execute:Y
Status:BLK3
CRC:ED82
```

Empfangen:

```
Block empfangen:
Status:ACK3
CRC:EA3E
```

Gesendet:

```
Status:TME3
```

```
CRC:F976
```

Empfangen:

```
Execute:Y
Status:BLK4
CRC:65E9
```

Gesendet:

```
Status:ACK4
CRC:EA39
```

Empfangen:

```
Status:EOT4
CRC:????

Status:EOT4
CRC:????

Status:EOT4
CRC:????
```

Ab hier werden die Daten nun gepackt - nur der angerufene brauchte Zeit. Nach dem Packen geht es nun weiter:

Empfangen:

```
Status:BEG5
CRC:????
```

Gesendet:

```
Status:EOT5
CRC:????

Status:EOT5
CRC:????

Status:EOT5
CRC:????
```

Hier wird nun das Vollduplex-Protokoll aufgerufen. Nach dem Aufruf des Protokolles bringt die Angerufene mit einem `NAK0`-Header den Block-Austausch wieder in Gang:

```
Status:NAK0
CRC:DA41
```

Gesendet:

```
Get:PB
Status:BLK1
CRC:9F7C
```

Empfangen:

```
Status:ACK1
CRC:EA3C
```

Gesendet:

```
Status:TME1
CRC:F974
```

Nach diesem Headeraustausch gelten die gesendeten und die empfangenen Daten als erfolgreich übertragen und können nach dem Logoff verarbeitet werden.

Empfangen:

```
Status:BLK2
Put:
CRC:6C61
```

Gesendet:

```
Status:ACK2
CRC:EA3F
```

Empfangen

```
Status:TME2
CRC:F977
```

Gesendet:

```
Execute:N
Logoff:Alle Daten ausgetauscht
Status:BLK3
CRC:ED82
```

Empfangen:

```
Status:ACK3
CRC:EA3E
```

Gesendet:

```
Status:TME3
CRC:F976
```

Empfangen:

```
Execute:N
Status:BLK4
CRC:65E9
```

Gesendet:

```
Status:ACK4
CRC:EA39
```

Empfangen:

```
Status:TME4
CRC:EA39
```

Und nun kann das Modem aufgelegt werden. In den ‘real existierenden’ ZCONNECT-Implementationen ist der Headeraustausch leider häufig nicht so klar, inbesondere werden häufig mehr Header übertragen, als eigentlich notwendig wären. Es wird auch keine Garantie für die Vollständigkeit des beschriebenen Datentransfers übernommen.

Kapitel III: Das Datenformat

In diesem Kapitel möchten wir nun beschreiben, was mit den komplett transportierten Daten passiert und wie diese aussehen. Betrachten wir aber zunächst, wie der gesamte Mechanismus zusammenarbeitet.

III.1 Vor dem Transport

In einem Server-System sammeln sich im Betrieb ständig Daten für die angeschlossenen Systeme. Je nach Philosophie der eingesetzten Software werden diese Nachrichten zwischengelagert (gespoolt) oder nur für den Transport vorgemerkt. Persönliche Nachrichten werden in der Regel zwischengelagert, da sie in der Server-MailBox selbst nicht einsortiert werden.

In regelmäßigen Abständen (z.B. alle 10 Minuten, alle halbe Stunde, jede Stunde oder - was aber nur bei Endsystemen sinnvoll ist - einmal täglich kurz vor 18:00 Uhr) werden diese Daten dann zum Transport freigegeben. Das dafür zuständige Programm sammelt entweder alle entsprechend markierten Daten aus der lokalen Datenbasis oder arbeitet das Zwischenlager (den Spool-Bereich) ab.

Dabei enstehen dann Netcall-Dateien für alle angeschlossenen Systeme, getrennt nach Privatmails und öffentlichen Nachrichten. Diese Netcalldateien werden nach ihrer Fertigstellung sofort mit dem für das System zuletzt eingestellten Packer an das bereits bereitliegende Netcall-Archiv angefügt.

Dabei werden Dateinamen benutzt, die *nur* aus folgenden Zeichen bestehen: die Ziffern von `0` bis `9` sowie den Großbuchstaben `A` bis `Z`. Es sind maximal acht Zeichen gefolgt von einem Punkt "`.`" und weiteren drei Zeichen erlaubt. Dies garantiert, daß die `ZCONNECT`- Archive auf allen Betriebssystemen ausgepackt werden können und beim Auspacken nicht Namenskollisionen entstehen (z.B. könnte ein UNIX-System die Dateien "`ArZk01`" und "`ARZK01`" einpacken, die dann auf einem DOS-System den gleichen Namen hätten).

Die Endung des Dateinamens (die drei weiteren Zeichen nach dem '`.`') kennzeichnen die Art der in einer Netcall-Datei enthaltenen Nachrichten. Hierbei gelten die folgenden Konventionen:

`*.brt`, `*.BRT`	Die Netcalldatei enthält ausschließlich öffentliche Nachrichten, d.h. Nachrichten, deren Empfängerin ein Brett ist.
`*.prv`, `*.PRV`	Die Netcalldatei enthält ausschließlich persönliche Nachrichten (PRV steht für *Privat*)
`*.kom`, `*.KOM`	Die Netcalldatei enthält Nachrichten, die öffentliche und private Empfänger gemischt enthalten können.
`*.eil`, `*.EIL`	Wie `*.KOM`. Die Nachrichten sind ausschließlich Eilmails.
`*.err`, `*.ERR`	Die Netcalldatei enthält ausschließlich nicht zustellbare (und deshalb zurückgeschickte) Nachrichten.
alle anderen	Die Netcalldatei kann alle Nachrichtentypen gemischt enthalten (wie `*.KOM`)

Unabhängig vom Dateinamen müssen selbstverständlich alle abgelieferten Dateien einsortiert werden.

Eine mögliche Implementierung der Namensgebung für die einzelnen Datenpakete im Archiv ist folgende: beim Zusammenstellen der Datenpakete wird die Zeit im Standard-UNIX-Format (Sekunden seit 1970) ermittelt. Diese wird dann in eine maximal achtstellige Hexadezimalzahl konvertiert und als Dateiname für das neue Datenpaket benutzt.[1]

III.2 Nach dem Transport

Angenommen, wir haben von einem anderen Netzwerksystem per ZCONNECT Daten empfangen. In der Regel befinden sich diese Daten alle gemeinsam in einem komprimierten Archiv. Der dazu verwendete Packer wird von ZCONNECT selbständig aktiviert, um die empfangenen Daten zu entpacken. Anschließend finden wir ein vorher leeres Verzeichnis auf unserer Festplatte, in dem sich alle empfangenen Dateien befinden. Minimal befindet sich hier überhaupt keine Datei (das andere System hatte keine Daten für uns), meist werden aber eine oder mehrere Dateien zum Einsortieren bereitstehen. Jede dieser Dateien kann aus mehreren Nachrichten bestehen. Die einzelnen Nachrichten stehen direkt, also ohne Trennzeichen, hintereinander. Üblicherweise befinden sich in einer Datei nicht gleichzeitig öffentliche und private Nachrichten dies ist jedoch nicht zwingend, jede Software muß daher in der Lage sein, auch gemischte Datenpakete einzulesen.

Jede Nachricht besteht aus zwei Teilen: dem *Header* und dem *Inhalt*. Wir werden diese beiden Teile getrennt diskutieren.

III.3 Inhalt

Der *Inhalt* kann dabei aus beliebigen Zeichen bestehen, er *muß* von jeder Software unverändert weitergegeben werden. Ausnahme: Gateways dürfen im Falle von Textnachrichten den Inhalt untersuchen und gegebenenfalls Umlaut- und Sonderzeichenkonvertierungen vornehmen.

Der Inhalt von Textnachrichten darf weder von der Software noch von der Systembetreiberin eingesehen bzw. interpretiert werden. Ausnahmen hiervon sind nur in technisch bedingten Notfällen (z.B. bei unzustellbaren Nachrichten) erlaubt, in diesem Fall

[1] Dabei muß gewährleistet sein, daß dies nicht öfter als einmal pro Sekunde geschieht...

muß die Empfängerin von der Kenntnisnahme durch die Systembetreiberin informiert werden. Es ist also nicht erlaubt, Textzeilen zum Steuern des Nachrichtenflusses zu mißbrauchen, z.B. ein "`To: ....`" in der ersten Zeile der Nachricht zur Adressierung zu verwenden. Alle Informationen, die für den Transport der Nachricht nötig sind und die durch den Transport entstehen ("Diese Nachricht wurde am xx.xx.xx um xx:xx Uhr von der geilsten MailBox in xxxx weitergeleitet"), sind im *Header* unterzubringen.

Nachrichten gelten als Textnachricht, wenn im Header die Information "`TYP:`" nicht enthalten ist. In diesem Fall besteht der Inhalt aus lesbaren Zeilen, die durch die Zeichenkombination `<CR> <LF>` (in dieser Reihenfolge) voneinander getrennt sind. Am Ende des Textes steht kein `<CTRL-Z>` als Datei-Ende-Zeichen (gehört zu den verbotenen Zeichen in Textnachrichten, siehe unten).

Nur Textnachrichten können von Userinnen gelesen werden, alle anderen Dateitypen erfordern bestimmte Anzeigeprogramme (Viewer), die in der Regel nicht online benutzt werden können (z.B. für Tondateien oder Grafiken). Die MailBox-Software konvertiert den unten definierten Standard-Zeichensatz in den von swe Userin benötigten Zeichensatz.

III.4 Header

Vor dem Nachrichten-Inhalt steht der Header. Er ist vergleichbar mit dem Briefumschlag der herkömmlichen Post. Die Informationen des Headers dürfen (und müssen sogar) zum Transport der Nachricht interpretiert und teilweise sogar verändert werden.

Der Header besteht aus beliebig vielen Informationen, gefolgt von einer Leerzeile (also der Zeichenfolge `<CR> <LF> <CR> <LF>`). Die einzelnen Informationen sind durch die Zeichenkombination `<CR> <LF>` getrennt.[2]

Jede *Information* besteht aus einer *Kennung* gefolgt von einem Doppelpunkt "`:`" und optional einem oder mehreren Leer- oder `<TAB>`-Zeichen. Anschließend folgt der eigentliche Informationsinhalt bis zum Ende der Zeile (`<CR> <LF>`). Die Länge der Zeile ist nicht begrenzt, sie darf von keiner Software beim Transport der Nachricht beschränkt werden.

Innerhalb des Informationsinhaltes sind alle Zeichencodes von 32- 255 erlaubt. Bei Informationsinhalten mit Text-Charakter (z.B. Betreff-Zeile) gelten die gleichen Zeichensatz-Einschränkungen wie für den Inhalt von Standard-Textnachrichten.

Die *Kennung* einer Information ist maximal 100 Zeichen lang. Sie besteht ausschließlich aus den Buchstaben `A-Z`, den Ziffern `0-9` sowie dem Bindestrich "`-`". Umlaute sind hier nicht erlaubt. Eine Kennung kann in Groß- oder Kleinschreibung oder auch beliebigen Kombinationen angegeben werden, die Analyse ist immer ohne Berücksichtigung der Groß-/Kleinschreibung vorzunehmen.

Im Header können Informationen in beliebiger Reihenfolge auftreten. Einzelne Kennungen können dabei auch mehrfach auftreten, bei anderen (z.B. `MID`, der Message-ID) ist das unsinnig und in diesem Fall explizit verboten.

Treten in einem eingelesenen Header Informationen mehrfach auf, ist die Reihenfolge dieser gleichen Informationen beizubehalten. Die Reihenfolge unterschiedlicher Informationen zueinander ist nicht definiert.

Einige Header sind Pflicht, das heißt, ohne sie wird die Nachricht, wenn möglich, zurückgeschickt. Das gilt auch für Nachrichten, bei denen Header mehrfach auftreten, die nur einmal erlaubt sind.

[2] Manchmal wird die Information auch mit 'Headerzeile' bezeichnet.

Dabei werden jedoch folgende Regeln beachtet:

- Ist ein ERR Header vorhanden, handelt es sich bereits um eine zurückgeschickte Mail. Diese wird auf keinen Fall zurückgeschickt, sie sollte direkt gelöscht werden, kann aber alternativ auch an die Systembetreuung weitergereicht werden.
- Ist (mindestens) ein EB Header vorhanden und ist dessen Adresse anders als die Absenderadresse, geht eine negative Empfangsbestätigung ("Nachricht nicht zustellbar. Grund: ....") sowie ein ERR Header an die in EB angegebene Adresse(n).
- Ansonsten geht die Nachricht an den Absender zurück. Zur Ermittlung der Adresse wird zunächst der WAB-Header, dann ANTWORT-AN und zuletzt der ABS-Header ausgewertet. Falls keine gültige Rücksendeadresse ermittelt werden kann, wird die Nachricht gelöscht.

Beim Zurückschicken wird die Fehlermeldung in einen ERR Header geschrieben und die Nachricht *samt Inhalt* zurückgeschickt!

III.5 Adressen

(siehe auch RFC-822) Netzadressen haben folgende Form:

¡lokaler-Teil¿@¡System-Name¿.¡Domain¿ (Vor- Nachname)

Hinter der eigentlichen Adresse (bis einschließlich ¡Domain¿) steht getrennt durch genau ein Leerzeichen in runden Klammern "(" ")" der zur Adresse gehörende Realname. Dieser Teil ist optional, wenn kein Realname angegeben wird, endet die Adresse mit der Domain.

Der Systemname und die Domain werden ohne Rücksicht auf Groß- / Kleinschreibung interpretiert. Ein System wird eindeutig durch eine Kombination aus Systemname und Domain beschrieben (d.h.: BIONIC.zer.de ist weltweit eindeutig), ein System kann aber mehrere Namen und Domains besitzen (z.B. BIONIC.comlink.de).

Beim Weiterleiten von Nachrichten genügt es nicht, nur den Systemnamen zu identifizieren. Beispielsweise gibt es im Z-Netz ein System namens "SOL". Schreibt nun eine Benutzerin der BIONIC eine Mail an "joe@sol.edu", so ist damit natürlich nicht die SOL im Z-Netz gemeint, sondern die in Californien. An die SOL im Z-Netz dürfen also nur Nachrichten mit Adressen wie "...@sol.ccc.de", "...@sol.zer.de" und Nachrichten ohne Domain-Angabe "...@sol" geschickt werden.

Empfängerangaben werden komplett an das Zielsystem weitergereicht, sie kommen also z.B. als "terra@sol.ccc.de" auf der SOL an. So kann die SOL entscheiden, ob sie selbst gemeint ist oder nicht, und gegebenenfalls die Nachricht für "joe@sol.edu" über den nächsten für ".edu" zuständigen Gateway (hier: die SOL selbst) weiterleiten.

Ist ein Empfängersystem dem lokalen System nicht bekannt (z.B. die sol.edu der BIONIC), leitet das System die Nachricht an den Domain-Server der angegebenen Domain (hier: .edu) weiter. Ist die Domain selbst nicht bekannt (z.B. .cs.ucb.edu) testet das System den Teil der Domain nach dem nächsten Punkt (hier: .ucb.edu) und gegebenfalls die Teile nach weiteren Punkten, bis eine bekannte Domain gefunden wird (hier: .edu).

In den einzelnen Teilen einer Adresse gelten unterschiedliche Beschränkungen des Zeichensatzes:

Systemname und Domain Hier sind nur die Buchstaben 'A' bis 'Z', die Ziffern '0' bis '9' sowie der Bindestrich '-' erlaubt.

lokaler Teil Hier sind alle Zeichen mit Codes von 33 '!' bis 124 '|' erlaubt, ausgenommen die Zeichen @<>/\()[]{}''",. Zeichen über 126, also auch die Umlaute, sind hier nicht gestattet. Die Zeichen '!' und '%' sind erlaubt, aber reserviert und dürfen daher nicht im Namen einer Userin auftreten.

Realname Hier sind alle ASCII-Zeichen von Leerzeichen (32) bis '~' (126) erlaubt,

lediglich die runden Klammern sind natürlich ausgenommen.

III.6 Brettnamen

Die andere Form der Adressierung ist ein Brettname: hier wird die Nachricht nicht an eine einzelne Empfängerin auf einem bestimmten System geschickt, sondern an alle Leserinnen eines öffentlichen Brettes.

Öffentliche Nachrichten enthalten kein '@' in der Empfängerangabe (und keinen Realnamen). Es ist möglich, eine Nachricht *gleichzeitig* an eine bestimmte Person und an ein öffentliches Brett zu schicken.

Für Brettnamen gelten folgende Regeln:

- Erlaubt sind die großen Buchstaben von 'A' bis 'Z', die Ziffern '0' bis '9', sowie '/', '_', '!', '+' und '-'. Auch hier sind Umlaute nicht erlaubt.
- Der Schrägstrich '/' dient zum Trennen von Brettern in Hierachieebenen. Ein Brettname beginnt immer mit einem '/', zwischen zwei '/' muß immer mindestens ein anderes Zeichen stehen. Ein Brettname endet nie mit einem '/'.

Der Unterschied zwischen einer öffentlichen Nachricht (Empfänger ist ein Brett) und einer persönlichen Nachricht besteht darin, daß öffentliche Nachrichten in der Regel kostenlos transportiert werden (also die Autorin nicht von der Systembetreiberin Gebühren berechnet bekommt), während persönliche Nachrichten auf Wunsch und auf Kosten der Absenderin versandt werden.

Eine denkbare Ausnahme hiervon ist z.B. ein Support-Brett, in dem Benutzerinnen von der Support-Anbieterin Gebühren für das Stellen von Fragen berechnet werden.

Eine persönliche Nachricht kann als Empfängerangabe auch ein Brett auf einem anderen System besitzen (`/EIN/BRETT@IRGENDWO.do.main`). Sie wird dennoch als persönliche Nachricht abgerechnet. Im Zielsystem wird diese Nachricht zunächst zensiert und muß von der Systembetreiberin freigeschaltet werden - falls das System einen derartigen Mechanismus unterstützt. Alternativ kann die Nachricht auch der Systembetreiberin zugestellt werden, die sie dann manuell in das gewünschte Brett transportiert. Ist sie dazu nicht bereit, schickt sie die Nachricht mit einer entsprechenden Fehlermeldung an die Absenderin zurück.

Dieser Mechanismus ist notwendig, da ansonsten die Schreibberechtigung in das gewünschte Brett nicht kontrolliert werden kann.

III.7 Weiterleiten

Beim Weiterleiten darf die Originalnachricht keinesfalls bearbeitet werden. Es ist lediglich erlaubt, einen Kommentar (mit Hilfe des `KOM` Headers) voranzustellen - die Nachricht selbst muß unverändert bleiben. Wird eine Nachricht manuell oder von einem Verteiler weitergeleitet, gibt es zwei Möglichkeiten: Die Absenderangabe wird ausgetauscht oder die Absenderangabe wird nicht ausgetauscht. In beiden Fällen werden folgende Aktionen vorgenommen:

- Die Nachricht bekommt eine neue Message-ID.
- Der Routweg im `ROT` Header wird gelöscht und leer neu hinzugefügt.
- Die ursprüngliche Empfängerin der Nachricht (kann auch ein Brettname sein) wird in den `OEM` Header kopiert, falls der `OEM` Header noch nicht existiert. Ansonsten bleibt der `OEM` Header unverändert.
- Wird an mehrere Empfängerinnen weitergeleitet, muß der KOP-Header (Kopienempfänger) entsprechend gesetzt werden. Im `KOP` Header sollten im Idealfall sämtliche Empfängerinnen, die diese Nachricht irgendwann erhalten haben, aufgeführt sein.
- Ein eventuell vorhandener `ERR` Header wird gelöscht.

- Ein eventuell vorhandener ZNETZ-Conf Header wird gelöscht.
- Falls der Header O-EDA nicht vorhanden ist, wird der Inhalt des EDA-Headers hier hinein kopiert. Anschließend wird in EDA das aktuelle Datum gesetzt.
- Der TRACE-Header wird gelöscht.
- Der VER-Header wird gelöscht.
- Der STAT-Header wird gelöscht.
- Der EB-Header wird gelöscht.

Die weitere Behandlung ist unterschiedlich. Falls die Absenderinnenangabe beim Weiterleiten ausgetauscht wird, wird wie folgt verfahren:

- Die Weiterleitende wird als Absenderin eingetragen, die absenderbezogenen Header ANTWORT-AN, MAILER, ORG, POST, PRIO, TELEFON werden gelöscht und mit den neuen Werten der Absenderin besetzt.
- Der WAB Header wird gelöscht.
- Die Absenderin der Originalnachricht wird in den OAB Header kopiert, falls noch kein OAB Header vorhanden ist. Als Absenderin wird der ANTWORT-AN-Header verwendet, wenn er vorhanden ist, sonst der ABS-Header. Falls der OAB-Header vorhanden ist, wird er nicht verändert.
- Eventuelle PGP-Unterschriften entfernen. (Header SIGNED und PGP-SIG)

Falls dagegen die Absenderangabe beim Weiterleiten beibehalten wird, wird weiterverfahren wie folgt:

- Die Weiterleitende wird in den WAB Header eingetragen, ein evtl. vorhandener WAB Header wird zuvor gelöscht.

Es gibt keine Möglichkeit, eine Nachricht weiterzuleiten, ohne die Message-ID zu ändern. Falls eine Systembetreiberin eine Nachricht in ein anderes Brett verschieben möchte, kann sie dies mit einem entsprechenden Befehl tun (im ZERBERUS z.B. "#copy"). Dies ist aber eine auf die lokale MailBox beschränkte Aktion, bei der nur folgendes geschieht:

- Das Brett, in das diese Nachricht verschoben wird, wird als EMP Header eingetragen
- Message-ID und Absenderangabe bleiben erhalten
- Der Routestring (ROT Header) bleibt erhalten

Diese verschobene Nachricht kann nicht mehr über das Netz transportiert werden, da sie die gleiche Message-ID behalten hat und daher von allen anderen Systemen (korrekterweise!) als Rekursion erkannt würde.

III.8 Automatisches Weiterleiten (Mailing-Listen, Netzwerk-Verteiler)

Eine MailBox oder auch eine externe Software kann die Möglichkeit bieten, über das Netz an einen automatischen Verteiler Nachrichten zu schicken. Dieser Verteiler verschickt jede Nachricht an alle eingetragenen Benutzerinnen dieses Verteilers.

Dieses Verfahren ähnelt einem öffentlichen Brett, allerdings ist nicht erforderlich, daß alle Systeme, über die die Nachrichten transportiert werden, das Brett führen. Dem gegenüber steigen natürlich die Kosten für eine solche Nachrichten-Verteilung, da alle Nachrichten, sowohl bis zum Verteiler als auch vom Verteiler zu den Benutzerinnen als PM verschickt werden.

Beim Eintreffen einer Nachricht in einem solchen Verteiler geschieht folgendes:

- Die Absenderin der Originalnachricht erhält eine Empfangsbestätigung - falls sie diese angefordert hatte (EB Header). Danach wird der EB-Header aus der Mail gelöscht.

- Alle Benutzerinnen des Verteilers werden ermittelt und als Empfängerinnen eingetragen.
- Die eingetragenen Kopienempfängerinnen bleiben erhalten! Es wird keine Kopienempfängerin hinzugefügt, dies geschieht erst beim Routen der Nachricht.
- Die Adresse des Verteilers wird in den OEM Header eingesetzt.
- Die Absenderin der Nachricht bleibt erhalten. Die E-Mail-Adresse der Betreuerin des Verteilers kann als WAB-Header eingesetzt werden. Dadurch gehen Antworten an den ursprünglichen Autor, Fehlermeldungen über falsche Einträge im Verteiler aber an dessen Verwalterin.
- Der TRACE-Header wird gelöscht.

III.9 Weiterleitungen durch Netzwerk-Vertreter

Bei Abwesenheit kann eine Benutzerin einer MailBox eine Vertreteradresse in ihrer MailBox angeben. Sämtliche Nachrichten, die ihr Postfach erreichen werden (auf ihre Kosten) automatisch an die dort angegebene Adresse weitergeleitet.

Beim Eintreffen einer Nachricht an eine Benutzerin, die einen Vertreter gesetzt hat, passiert folgendes:

- Enthält die Nachricht eine Empfangsbestätigung, so wird diese nicht versendet. Der Empfang wird durch die Vertreterin bestätigt.
- Da es sich um eine persönliche Nachricht handelt, braucht die Message-Id bei dieser Art der Weiterleitung nicht geändert zu werden.
- Die Empfängeradresse (EMP-Header) wird in den VER-Header kopiert.
- Der Header O-ROT wird gelöscht und mit dem Routstring (Header ROT) der Nachricht gefüllt.
- Danach wird der HeaderROT geleert, d.h. der Header wird gelöscht und es wird der Name der weiterleitenden MailBox (mit Domain) eingetragen.

III.10 Beispiel

Beispiel für eine ZCONNECT-Nachricht (von Martin Husemann auf dem Point MARTIN der BIONIC an Martin Husemann auf dem System “sisyphus.owl.de”):

```
EDA:  19920607140703S+2
BET:  Dies ist ein Routingtest
MID:  70.54215@MARTIN.BIONIC.zer.de
ABS:  M.Husemann@BIONIC.zer.de (Martin Husemann)
ROT:  BIONIC.zer.de
EB:
EMP:  M.Husemann@sisyphus.owl.de
LEN:  103
PRIO: 0

Hallo Martin,
falls Du das hier über BI-LINK bekommst,
stimmt das Routing.

Gruß, Martin
```

III.11 Mögliche Header-Informationen

ABS Absenderin, Pflicht, nur einmal

Die Adresse, über die die Absenderin erreichbar ist, komplett mit Absendersystem, Domainangabe und evtl. Realname.

ANTWORT-AN optional

Eine private Antwort an die Absenderin ist nicht an die ABS-Adresse zu schicken, sondern an die hier angegebene. Dies ermöglicht Benutzerinnen mehrerer Mail-Boxen, alle Antworten an die "Hauptadresse" schicken zu lassen. Auch bei automatisch generierten Nachrichten (Absenderin "Mailer-Daemon") kann so eine Ansprechpartnerin für Rückfragen angegeben werden.

BET Betreff, Pflicht, nur einmal

Ein bei Antworten automatisch generierter Betreff ist so zu wählen, daß vor den Betreff der Originalnachricht ein 'Re:␣' gesetzt wird, falls dort nicht bereits 'Re:␣' oder 'RE:␣' steht. Ansonsten wird der Betreff unverändert übernommen.

Es gibt also nur "Re: xxx", niemals aber "Re: Re: xxx". Die Verschachtelungstiefe der Antworten ist aus den BEZ Headern zu entnehmen.

BEZ Bezug, optional, mehrfach

Wenn diese Nachricht eine Antwort auf eine ältere Nachricht ist, gibt der Bezug die Message-ID der Originalnachricht an. Wenn mehrere Bezüge enthalten sind, so stehen ältere vor neueren.

Stellt eine zu erzeugende Nachricht eine Antwort in welcher Form auch immer (automatisch, manuell, Fehler, etc.) dar, so ist die Message-ID der Nachricht, auf die geantwortet wird, in den Header BEZ zu setzen. Enthielt die ursprüngliche Nachricht bereits einen oder mehrere BEZ, so ist der neue Bezug als letzter an die bereits vorhandenen Bezüge anzuhängen. Die Reihenfolge der bisherigen Bezüge darf nicht verändert werden.

CHARSET Zeichensatz, optional, nur einmal

Mit diesem Header kann definiert werden, welcher Zeichensatz in einer Textnachricht verwendet wird. Mögliche Werte sind:

ISO1, ISO2, ISO3, ISO4, ISO5, ISO6, ISO7, ISO8, ISO9, UNICODE.

Hierbei meint ISO1 bis ISO9 den Zeichensatz ISO-8859-1 bis -9. Es wird dringenst empfohlen, mindestens ISO1 zu unterstützen.

Hinweis: Ist der Header CHARSET nicht vorhanden, so gilt aus Kompatibilitätsgründen die Zeichensatzdefinition aus ZCONNECT Version 3.0.

Der Header CHARSET wirkt sich nur auf den Body der Nachricht aus, niemals auf den Header. Ein Nachrichten-Header ist immer im ASCII-7-Bit-Format.

Der Parameter im Header ist nicht case sensetiv.

CRYPT Crypter, optional, nur einmal

Kennzeichnet diese Nachricht als verschlüsselt. Dieser Header enthält ein Schlüsselwort, das das Verschlüsselungsverfahren angibt. Folgende sind bisher definiert:

PMCRYPT2	Ein von XPoint verwendetes Verfahren
DES/ECB	NSA LowTech: Electronic Code Book
DES/CBC	DES Cipher Block Chain
DES/CFB	DES Cipher Feedback
DES/OFB	DES Output Feedback
PGP	Pretty Good Privacy
QPC	QuickPoint Crypt

Die Schlüsselworte sind unabhängig von Groß/Kleinschreibung auszuwerten.

CRYPT-CONTENT-TYP optional, nur einmal

Verschlüsselte Nachrichten sind immer Binärnachrichten. Der ursprüngliche Typ der Nachricht wird hierher übernommen.

CRYPT-CONTENT-KOM optional, nur einmal

Es wird immer alles verschlüsselt und unterschrieben, was unter LEN: steht, also inklusive eines eventuelle Kommentars. Damit der korrekte Kommentar wieder rekonstruiert werden kann, wird KOM: beim Verschlüsseln in CRYPT-CONTENT-KOM umbenannt. Selbstverständlich kann danach noch einmal ein Kommentar angehangen werden, zum Beispiel beim Weiterleiten einer fehlerhaften Nachricht.

DDA Dateidatum, optional, nur einmal

Gibt das Datum der letzten Änderung einer Datei an. Das Format des Datums ist unter EDA beschrieben.

DISKUSSION-IN optional, auch mehrfach

Gibt die Empfängerin an, die bei öffentlichen Antworten benutzt werden soll. Dies ist immer dann sinnvoll, wenn eine Nachricht in mehrere Bretter geschickt wird, die darauf folgende Diskussion aber auf ein Brett beschränkt werden soll. Es können aber auch reine Informationsbretter von Diskussionsbeiträgen freigehalten werden, indem die Antworten auf ein passendes Diskussions-Brett dirigiert werden.

Im Header DISKUSSION-IN können auch E-Mail-Adressen gesetzt werden. Das absendende System sollte dann aber sicherstellen, daß hier nur E-Mail-Adressen der Absenderin eingesetzt werden können, um Mißbrauch keinen Vorschub zu leisten.

EB optional, auch mehrfach

Ist dieser Header vorhanden, verschickt das Zielsystem, sobald die Nachricht von ihm empfangen wird, eine Empfangsbestätigung an die Absenderin. Benutzt die Empfängerin einen Point und ist dieser auch mit ZCONNECT angeschlossen, wird die Empfangsbestätigung nicht beim Empfang in der MailBox ausgelöst, sondern erst vom Point. In allen anderen Fällen wird beim Einsortieren der Nachricht in die MailBox sofort die Bestätigung verschickt. Bestätigt wird der Empfang, nicht das Lesen der Nachricht (Datenschutz!).

Der EB Header kann auch eine Adresse enthalten, in diesem Fall geht die Empfangsbestätigung nicht an die Absenderin, sondern an die angegebene Adresse. Sind mehrere EB Header vorhanden, erhält jede dort aufgeführte Adresse eine Bestätigung.

In der Bestätigung ist im BEZ Header die Message-ID der bestätigten Nachricht anzugeben. Weiterhin ist ein Header STAT: EB zu setzen.

EDA Datum, Pflicht, nur einmal

Das Erstellungsdatum wird dabei im Format JJJJMMTThhmmss[S/W](+/-offset) angegeben, wobei S oder W für Sommer bzw. Winterzeit steht, offset ist die Zeitzone als Unterschied in Stunden zur GMT. Dabei wird die Zeit immer als GMT angegeben, die Zeitzone / Sommerzeit ermöglicht lediglich das Umrechnen dieser universellen Zeit auf die lokale Zeit der Absenderin. In Deutschland gelten die Zeitzonen MET und im Sommer MEST. Diese würden durch die Zusätze “W+1” bzw. “S+2” dargestellt Durch die Darstellung als GMT sind die JJJJMMTThhmmss Angaben auch während der Umstellung von Sommer- auf Winterzeit und umgekehrt kontinuierlich.

Falls die lokale Uhrzeit nicht um ganze Stundenbeträge von GMT abweicht, wird dem Offset eine Minutenangabe zugefügt. Beispiel: “W-9:30” für die Zentral-Australische-Zeitzone.

Siehe auch Anhang “Zeitzonen”.

EMP Empfänger, Pflicht, mehrfach

Die Netzadresse der Empfängerin(nen).

Tritt diese Information mehrfach auf, muß diese Nachricht an jede dieser Empfängerinnen weitergeleitet werden. Geschieht dies nicht über ein gemeinsames Routesystem, sind Kopien der Nachricht anzufertigen.

Bei diesem Kopiervorgang bekommen die einzelnen Kopien nur noch die EMP Header, an die diese Kopie weitergehen soll. Alle übrigen Empfänger (die über ein anderes System erreicht werden sollen) werden als Kopienempfänger (KOP Header) eingetragen, falls nicht STAT:NOKOP angegeben ist.

Enthält eine der EMP-Angaben keinen "@", handelt es sich um eine öffentliche Nachricht. Eine Nachricht kann in mehrere öffentliche Bretter geschickt werden, indem für jedes Brett eine EMP-Information eingesetzt wird. Physikalisch wird natürlich nur eine Kopie der Nachricht weitergereicht. Hier wird also - im Gegensatz zu den privaten Nachrichten - niemals kopiert.

Hat ein System nicht alle der in EMP Headern angegebenen Bretter bestellt, müssen dennoch alle EMP Header weitergegeben werden! Das gleiche gilt, wenn auf dem lokalen System nicht jedes Brett, das in einem EMP Header aufgeführt wird, existiert.

Ein EMP Header darf auch einen Realnamen enthalten.

ERR Error, optional, nur einmal

Falls dieser Header vorhanden ist, wurde die Nachricht von einem Programm automatisch zurückgeschickt - entweder weil die Nachricht fehlerhaft oder die Empfängerin unbekannt war. Der Inhalt des ERR-Headers ist i.W. die Fehlermeldung im Klartext. Optional kann der Fehler vor der Meldung durch eine Zahlenfolge identifiziert werden:

```
ERR: [ErrClass[;ErrNo[;...]]] [ErrMessage]
```

Ein leerer ERR-Header ist unzulässig.

ErrMessage enthält einen nach Möglichkeit englischsprachigen Text, der den Fehler möglichst treffend beschreibt. ErrMessage ist nur dann optional, wenn mindestens ErrClass definiert ist.

ErrClass enthält die Fehlerklasse. Dabei sind Werte ≥ 0 möglich. Werte im Bereich $0 \leq$ ErrClass ≤ 4 signalisieren hierbei, daß die Nachricht trotz Fehlern zugestellt worden ist; Der Bereich $5 \leq$ ErrClass ≤ 9 signalisiert, daß die Nachricht nicht zugestellt werden konnte.

Für ErrNo gilt folgende Belegung:

1	Versand nicht möglich.
1;1	Konto überzogen.
1;2	Mail zu alt.
1;3	Netzzugriff für Absenderin gesperrt.
2	PM-Zustellung nicht möglich.
2;1	Keine Empfängerin der Nachricht angegeben.
2;2	Empfängerin im Zielsystem unbekannt.
2;3	Verteiler schreibgeschützt.
3	PM-Routing nicht möglich.
3;1	Routsystem unbekannt oder gesperrt.
3;2	Direktmails & Routmails gesperrt.
3;3	Routing nicht möglich da Mail zu lang.
3;4	System in Domainserver unbekannt.
3;5	Eingestellter Domainserver unbekannt.
3;6	PM-Rekursion.

3;7	Empfangssystem gesperrt.
3;8	Domain unbekannt.
4	Brett-Zustellung nicht möglich.
4;1	Brett für Netcallempfang gesperrt.
4;2	Brettname unzulässig.
4;3	Brett existiert nicht.
4;4	Brett gesperrt.
4;5	Kein Autoeintrag; mehrfache Brettangaben.
5	Verletzung von Regeln für den Header.
5;1	Ein nur einmal erlaubter Header tritt mehrfach auf.
5;2	Ein Pflichtheader ist nicht vorhanden.
5;3	Ein Header hat falsches Format.
5;x;1	`ABS`-Header, $1 \leq x \leq 3$, siehe oben.
5;x;2	`EMP`-Header, $1 \leq x \leq 3$, siehe oben.
5;x;3	`EDA`-Header, $1 \leq x \leq 3$, siehe oben.
5;x;4	`BET`-Header, $1 \leq x \leq 3$, siehe oben.
5;x;5	`ROT`-Header, $1 \leq x \leq 3$, siehe oben.
5;x;6	`GAB`-Header, $1 \leq x \leq 3$, siehe oben.
5;x;7	`MID`-Header, $1 \leq x \leq 3$, siehe oben.
5;x;8	`WAB`-Header, $1 \leq x \leq 3$, siehe oben.
5;x;9	`KOP`-Header, $1 \leq x \leq 3$, siehe oben.
5;x;10	`OAB`-Header, $1 \leq x \leq 3$, siehe oben.
5;x;11	`OEM`-Header, $1 \leq x \leq 3$, siehe oben.
5;x;12	`EB`-Header, $1 \leq x \leq 3$, siehe oben.
5;x;13	`Antwort-An`-Header, $1 \leq x \leq 3$, siehe oben.
5;x;14	`Diskussion-In`-Header, $1 \leq x \leq 3$, siehe oben.

`ERSETZT` optional

Gibt die Message-ID der Nachricht an, die von dieser ersetzt wird. Damit kann dafür gesorgt werden, das von einer regelmäßig veröffentlichten Information immer nur die aktuelle Version in einer MailBox vorhanden ist. Anwendungsbeispiele: Serverstruktur, MailBox-Listen, ZMAPs, FAQs etc. Falls der Nachrichtentext der Nachricht leer ist, so ist die betreffende “ersetzte” Nachricht zu löschen.

Beide Anwendungsfälle, Nachrichtenersetzung und netzweite Rücknahme bedürfen einer Prüfung der Autorisierung. Dies kann beispielsweise durch die Prüfung der Identität der Absenderin oder mit Hilfe des öffentlichen Schlüssels der Absenderin geschehen.

Bei der Implementation des `ERSETZT`-Headers ist es ausreichend, wenn bei der alten Nachricht notiert wird, daß diese inzwischen ersetzt wurde bzw. vom Autor zurückgezogen wurde.

`FILE` Filename, optional, nur einmal

Gibt den Dateinamen (ohne Directory!) der Datei an - zum Beispiel für Binärnachrichten oder Grafiken.

! → Je nach Betriebssystem des Absende-Systems, kann dieser Dateiname beliebig lang sein und evtl. Sonderzeichen, Leerzeichen sowie natürlich mehrere Punkte enthalten! Jede Software, die diesen Header auswertet, um diese Nachricht zu speichern, sollte darauf vorbereitet sein und entweder den Namen entsprechend kürzen sowie ungültige Zeichen durch Ersatzdarstellungen ersetzen oder bei ungültigen Namen einen eigenen Namen generieren.

`KOM` Kommentarlänge, optional, nur einmal

Länge des Kommentars in Byte. Wird z.B. für Binärnachrichten, denen ein ASCII-Kommentar vorangestellt ist, gebraucht. Nach dem Header folgt der Kommentar in der angegebenen Länge, dann erst die Binärdaten. Die Binärdatenlänge ist also LEN minus KOM. Ein Kommentar kann aber auch bei allen anderen Nachrichtentypen vorangestellt werden. Für den Inhalt des Kommentars gelten immer die Regeln für Standard-Textnachrichten, auch wenn er einem Text mit alternativem Zeichensatz (und entsprechender TYP- Information) vorangestellt ist.

KOP Kopienempfängerin, optional, mehrfach

Falls eine Nachricht an mehrere Personen geschickt wurde, kann diese Information die übrigen Empfängerinnen auflisten. Gibt es mehrere Kopienempfängerinnen, tritt diese Information mehrfach mit jeweils einer Empfängeradresse auf (je eine KOP-Information pro Empfängerin).

! → Diese Information dient nur der Dokumentation für die Empfängerinnen, sie wird nicht zum Steuern der Nachrichtenweiterleitung verwendet. Falls eine KOP Angabe gemacht wird, aber keine entsprechende EMP Angabe vorhanden ist, wird die routende Software sich nicht bemühen, dieser KOP-Adressatin eine Kopie zuzusenden. Die Software wird vielmehr davon ausgehen, daß diese Adressatin ihre Kopie bereits über einen anderen Routweg erhalten hat (bzw. erhalten wird).

LANGUAGE Antwortsprache, optional, nur einmal

Mit diesem Header kann die Absenderin einer Nachricht angeben, in welcher Sprache sie gerne ihre Antworten erhalten möchte. Dieses sollte auch und gerade bei automatischen Nachrichten berücksichtigt werden.

LANGUAGE enthält einen Parameter. Dabei handelt es sich um den englischsprachigen Namen der betreffenden Sprache, also GERMAN für Deutsch. Im folgenden eine Liste verwendbarer Kürzel im LANGUAGE-Header:

Kürzel	Sprache
GERMAN	Deutsch
ENGLISH	Englisch
SPANISH	Spanisch
FRENCH	Französisch
GREEK	Grichisch

Kann eine Sprachpräferenz nicht befriedigt werden, so sollte nach Möglichkeit Englisch verwendet werden. Ist der Header nicht vorhanden, kann die Antwort aus Kompatibilitätsgründen in Deutsch erfolgen.

LDA Löschdatum, optional, nur einmal

Ein Datum, ab dem diese Nachricht automatisch gelöscht werden soll / kann. Kann für Veranstaltungshinweise oder andere Nachrichten mit “Verfallsdatum” (z.B. die urgent actions von amnesty international) verwendet werden.

LEN Länge, Pflicht, nur einmal

Die Länge des Inhalts (alles, was hinter dem Header noch zu dieser Nachricht gehört) in Byte. Auch die Länge 0 ist erlaubt.

MAILER Mailer, optional, nur einmal

Gibt den Namen des (von der Absenderin, bzw. vom konvertierenden Gateway) verwendeten Mailers an. (pure Werbung, aber immerhin für Userinnen unsichtbar) Dient der Fehlererkennung im Netzwerk. Hier sollte eine eindeutige Kennung der Software incl. Versions- und Releasenummer stehen.

MID Message-ID, Pflicht, nur einmal

Die Message-ID muß wie eine gültige Adresse (ohne Realname) aussehen (siehe

oben) und darf innerhalb von zwei Jahren weltweit nicht wiederholt werden. Dazu müssen Message-IDs eine Domain enthalten.

Die Message-ID dient zur eindeutigen Identifikation dieser Nachricht. Sollte innerhalb von zwei Jahren eine Nachricht mit einer gleichen Message-ID noch einmal auftreten, ist dies eine *Rekursion*, d.h. die Nachricht ist über einen Umweg noch einmal zur MailBox gelangt und kann deshalb gelöscht werden. Sie darf auf keinen Fall weitergeleitet werden.

Eine praktische Implementationsmöglichkeit ist es z.B., alle Message-IDs für 90 Tage aufzubewahren und alle eingehenden Nachrichten gegen diese Datenbank zu prüfen. Eingehende Nachrichten, die älter als 90 Tage sind, können bedenkenlos entsorgt werden, ohne die Message-ID zu testen.

Der Rekursionstest anhand der Message-ID muß von jeder Software durchgeführt werden! Öffentliche Nachrichten, die als Rekursion erkannt wurden, dürfen nicht weitergeroutet werden.

Persönliche Nachrichten werden nicht auf Rekursion geprüft, lediglich das Zielsystem der Nachricht darf doppelte persönliche Nachrichten ausfiltern.

In Message-IDs sind die Zeichen ‘<’, ‘>’ und ‘/’ verboten.

`O-ROT` Original-Routweg, optional, nur einmal

Für diesen Header gelten dieselben Formatregeln, wie für den Header `ROT`.

Bei Weiterleitungen durch Netzwerkvertreterinnen (siehe Abschnitt III.9 wird der Routstring (Header `ROT`) in diesen Header kopiert. Hierdurch wird bei persönlichen Nachrichten das fälschliche Melden von Ping-Pong-Routing verhindert.

`O-EDA` Original-Erstellungsdatum, optional, nur einmal

Dieser Header enthält ein Nachrichtendatum. Es gelten die Regeln für Datumsangaben, die bei Header `EDA` erläutert wurden.

Um das versehentliche Löschen von weitergeleiteten Mails als "zu alt" zu verhindern, muß bei Weiterleitungen ein aktuelles Nachrichtendatum eingesetzt werden. Damit das Original-Erstellungsdatum nicht verloren geht, wird es bei der ersten Weiterleitung der Nachricht (`O-EDA` existiert nicht) in diesen Header kopiert.

`OAB` Original-Absenderin, optional, nur einmal

Falls eine Nachricht manuell oder per Verteiler weitergeleitet wurde, steht hier, wer die Nachricht original verschickt hat.

`OEM` Original-Empfängerin, optional, mehrfach

Falls eine Nachricht manuell oder per Verteiler weitergeleitet wurde, steht hier die ursprünglich angegebene Empfängerin.

`ORG` Organisation, optional, nur einmal

Eine kurze, einzeilige Beschreibung der hinter der Absenderin stehenden Organisation, z.B. "Borland Deutschland GmbH, Starnberg, F.R.G.". Wird eine solche Information eingesetzt und die Nachricht gibt nicht die offizielle Meinung der Organisation wieder, wird im Nachspann (Signatur) der Nachricht meist der "Standard- Disclaimer" eingefügt: "Meine Meinung ist *nur* meine Meinung. Sie wird von meiner Arbeitgeberin weder geteilt noch bezahlt."

`PGP` optional, mehrfach

Enthält weitere Informationen bezüglich PGP-Verschlüsselung. Diese Zeile kann auch bei nicht mit `PGP` bearbeiteten Nachrichten auftreten. Derzeit mögliche Werte:

`PLEASE` Die / der AbsenderIn bittet die / den EmpfängerIn, den im Header `PGP-PUBLIC-KEY` erhaltenen Public-Key (zur AbsenderInnenadresse) zu verifizieren. Gilt nicht in öffentlichen Nachrichten.

Das Pointprogramm sollte darauf aufmerksam machen, daß die Verifizierung eines Schlüssels nur durch persönlichen Kontakt erfolgen sollte, also zum Beispiel ein Telefongespräch oder eine direkte Begegnung mit Abgleich des Fingerprints.

`REQUEST` Fordert das Pointprogramm auf, den "eigenen" Public-Key als Mail an die AbsenderInnenadresse zu schicken. Gilt nicht in öffentlichen Nachrichten.

`PGP-ID` optional, nur einmal

Zu jedem PGP-Public-Key gehört eine User-ID, die sich normalerweise aus Netzadresse und Realnamen zusammensetzt, in der Form `Real Name <Netzadresse>` zusammensetzt. Zum Beispiel:

```
Christoph Teuber <ch_teuber@aworld.aworld.de>
```

Weicht die PGP-UserId von dem aus der Adresse ermittelbaren Header ab, so kann die richtige ID in diesem Header transportiert werden.

`PGP-PUBLIC-KEY` optional, nur einmal

Enthält den PGP-Public-Key des Absenders. Genaueres siehe Abschnitt III.13.2.

`PGP-KEY-AVAIL` optional, auch mehrfach

Kann in jeder Nachricht stehen und signalisiert, daß die / der AbsenderIn die Nutzung von PGP anbietet und wie der öffentliche Schlüssel zu beziehen ist.

Grundsätzlich, also auch bei leerer Headerzeile, ist der zur AbsenderInnenadresse gehörige öffentliche Schlüssel über eine private Nachricht mit gesetztem `PGP:REQUEST` zu erhalten.

Zusätzlich kann dort eine Zeile im Format

```
+49-5202-88888 bi-link.owl.de martin@bi-link.owl.de
```

Der Public-Key kann dann per ZCONNECT Key-Request abgeholt werden. Parameter:

1. Telefonnummer im internationalen Format einer Modem-Leitung dieser Mail-Box.
2. Systemname der MailBox, die den Key-Server betreibt. Wird weggelassen, wenn dies dem System der AbsenderInnenadresse entspricht.
3. UserInnenname, die Netzadresse, die im Key angegeben ist. Dies wird im `PGP-KEYREQ` Header angegeben. Wird weggelassen, wenn dies der AbsenderInnenadresse entspricht.
4. Optionaler Parameter `ISDN`, falls es sich bei der Telefonnummer um einen ISDN-Anschluß handelt.

`PGP-KEY-COMPROMISE` optional, nur einmal

Inhalt genau wie `PGP-PUBLIC-KEY`. Dieser Header enthält einen widerrufenen Key. Er darf nur zusammen mit dem neuen Key (als `PGP-PUBLIC-KEY`) auftreten.

`PGP-KEY-OWN` optional, nur einmal

Inhalt genau wie PGP-PUBLIC-KEY: Enthält den eigenen Schlüssel mit einer neuen Unterschrift. Dies sollte als Antwort auf ein `PGP:PLEASE` generiert werden.

`PGP-SIG` optional, nur einmal

Falls die Nachricht mit PGP unterschrieben ist – der Header `SIGNED:PGP` ist vorhanden – steht in diesem Header die BASE64-Codierung der PGP-Unterschriftsdatei. Diese Datei läßt sich mit `PGP -sb` erzeugen und analog zum Header `PGP-PUBLIC-KEY` in einen Header umwandeln.

`POST` Post-Adresse, optional, nur einmal

Wenn die Absenderin einer Nachricht auch über andere Medien, z.B. per Post, erreichbar sein möchte, kann sie in diesem Header ihre postalische Anschrift unterbringen. Die einzelnen Anschriftenzeilen werden hintereinander geschrieben und jeweils durch Semikola “`;`” getrennt.

`PRIO` Priorität, optional, nur einmal

Ist dieser Headerinformation nicht angegeben, gilt `PRIO:0`. Gibt die Dringlichkeit der Zustellung an. Zur Zeit sind folgende Dringlichkeiten definiert:

0	normal (per Routing)
10	direkt zum Zielsystem
20	Eilmail (direkt mit sofortiger Auslieferung)

Nachrichten, die über den Routweg bei einem Serversystem eintreffen, dürfen jedoch immer auch per Routing weiterverteilt werden.

`ROT` Routweg, Pflicht, nur einmal

Jedes System trägt sich hier ein, wenn es die Nachricht empfängt. Eine Nachricht (auch eine PM) darf niemals an ein System weitergereicht werden, dessen Name bereits im Routweg steht.

Falls eine PM über ein System zugestellt werden muß, das bereits im Routweg steht, sollte diese Nachricht dem Sysop vorgelegt werden (Achtung: Datenschutz! Nur die Header, nicht der Nachrichteninhalt darf sichtbar sein!), da offenbar ein Ping-Pong-Routing besteht.

Als erstes System trägt sich hier das Absender-System ein (damit auch dieses die Nachricht nicht noch einmal bekommt). Erreicht die Nachricht das nächste System, setzt dieses seinen eigenen Namen (incl. Domain) gefolgt von einem '!' vor den alten Inhalt dieser Information. Dazu ein Beispiel: auf der BIONIC.zer.de wird eine Nachricht erzeugt:

```
ROT: BIONIC.zer.de
```

Nun erreicht diese Nachricht die BI-LINK.owl.de:

```
ROT: BI-LINK.owl.de!BIONIC.zer.de
```

SIGNED optional, nur einmal

Enthält Informationen darüber, wie und ob die Nachricht authentisiert wurde. Derzeit mögliche Werte

PGP Die Nachricht ist mit PGP unterschrieben. Dies kann zusätzlich zu einer Verschlüsselung erfolgt sein (CRYPT:PGP ist dann ebenfalls gesetzt) oder einzeln.

Falls die Nachricht nicht zusätzlich verschlüsselt ist, steht die Unterschrift in PGP-SIG.

SPERRFRIST Gültig ab, optional, nur einmal

Ein Datum im Format wie EDA. Vor diesem Datum wird diese Nachricht *nicht* angezeigt. Damit kann z.B. eine Sperrfrist bei Pressemeldungen eingehalten werden.

STAT Nachrichtenstatus, optional, mehrfach

Beschreibt, was die Nachricht ist: Falls dieser Header fehlt, handelt es sich um eine normale Mail. Wenn der Header vorhanden ist, gibt es folgende Einträge:

AUTO	Nachricht ist eine automatisch und regelmäßig versandte Nachricht. Diese Information kann genutzt werden, um die frühere Version zu löschen.
EB	Nachricht ist eine automatisch verschickte Empfangsbestätigung.
CTL	Nachricht ist eine Kontrollnachricht, die - auch wenn sie defekt ist - nicht zurückgeschickt werden darf.
TRACE	Kennzeichnet eine Nachricht als TRACE-Nachricht. Beschreibung, siehe TRACE Header.
NOKOP	Bestimmt, daß beim Routen EMP-Header nicht in KOP-Header kopiert werden sollen. (siehe EMP-Header)
NOCIPHER	Bestimmt bei nicht verschlüsselten Nachrichten, daß auch eventuelle Antworten darauf nicht automatisch verschlüsselt werden.

Die Schlüsselworte sind unabhängig von Groß / Kleinschreibung auszuwerten.

STICHWORT Stichworte, optional, mehrfach

Listet einzelne Stichworte zum Inhalt der Nachricht. Jeder STICHWORT-Heder

enthält nur ein Stichwort.

Die Suche nach Stichworten ist unabhängig von der Groß/Kleinschreibung. Es sind nur die Zeichen von A-Z in diesem Header zugelassen.

TELEFON Telefonnummer, optional, nur einmal

Hier kann die Absenderin ihre Telefonnummer(n) unterbringen. Es wird die internationale Schreibweise verwendet, mit vorangestelltem "V" für Voice, "F" für Fax, "P" für Pager (Cityruf) oder "B" für MailBox (BBS). Bei Voice-Nummern wird ein "Q" nachgestellt, wenn ein Anrufbeantworter vorhanden ist. Alle Nummern werden durch Semikolon oder Leerzeichen getrennt. Beispiel:

```
TELEFON: V+49-521-561345Q  F+49-521-561785
         B+49-521-193004
```

Bei kombinierten Nummern werden die Kennbuchstaben hintereinandergestellt.

TRACE optional, nur einmal

Enthält eine E-Mail-Adresse, an die die Trace-Information gesendet werden soll. Jedes System, das eine so gekennzeichnete Nachricht empfängt (bzw. weiterleitet) sendet eine Trace-Meldung an die angegebene Adresse. Die Trace-Meldung enthält (u.a.) die folgenden Header:

BEZ: (orig-id)
BET: Trace-Info (orig-id)
STAT: CTL
STAT: TRACE

Der Inhalt ist eine Liste, an welche Systeme diese Nachricht mit welchen EMP: Headern weitergeleitet wurde. Um den Inhalt sprachunabhängig parsbar zu machen, werden alle Systemnamen in < > gesetzt. Für jedes Routsystem gibt es einen Abschnitt bestehend aus der Identifikation dieses Systems (also <name>) gefolgt von einer Liste der zu diesem System geschickten EMP Header. Die EMP Liste enthält einen EMP pro Zeile und beginnt mit einem Leerzeichen oder Tabulator. Sobald eine Zeile wieder direkt am Zeilenanfang beginnt, ist dies die nächste Routsystem-Zeile.
Beispiel fuer eine Antwort-Mail:

```
An <bionic.owl.de> gingen folgende EMP: Header:
   <sysop@link-goe.central.de>
   <padeluun@bionic.zer.de>
   <ghostwriter@hot.zer.de>
An <hsp.zer.de> gingen folgende EMP: Header:
   <h.schulze@hsp.zer.de>
An <bht-box.zer.de> gingen folgende EMP: Header:
   <sysop@bht-box.zer.de>
```

Die Trace-Information darf nur bei persönlichen Nachrichten stehen (und auch nur dann beantwortet werden). Der TRACE-Header ist bei Weiterleitungen zu löschen.

TYP Typ, optional, nur einmal

Nähere Beschreibung des Dateityps. Definiert, um welche Art von Binärdatei es sich handelt (z.B. TIFF, GIF, PCX, ...). Alle unbekannten TYP Informationen werden als reine Binärnachricht aufgefaßt. Definiert sind die Typkennungen:

BIN	allgemeine Binärnachricht
TRANSPARENT	Textnachricht ohne Umlautwandlung
RFC1563	MIME-Subtype Text/Enriched von N. Borenstein aus RFC 1563
MIME	Mime nach RFC 1341. Siehe auch Anmerkungen weiter unten.

Beim Inhalt des TYP Headers wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Falls kein TYP-Header vorhanden ist, handelt es sich um eine Textnachricht.

Hinweis: Die Kennungen RFC1563 und MIME... waren bei Drucklegung dieses Dokuments noch nicht beschlossen. Sie sind hier der Vollständigkeit halber mit beschrieben. Bitte lesen Sie genaueres in der README-Datei der beigefügten Diskette.

Falls der Typ MIME spezifiziert ist, wird er folgendermaßen in ZCONNECT integriert. Der Mime-Header MIME-Version wird bedeutungsgleich durch den ZCONNECT-Header MIME dargestellt. Für die anderen Mime-Header gelten folgende Namen:

MIME-Header	Name in ZCONNECT
Content-Type	MIME-Type
Content-Transfer-Enconding	MIME-Encoding
Content-ID	MIME-ID
Content-Description	Zusammenfassung

Diese Namens-Umwandlung betrifft nur die Mime-Header, die innerhalb des ZCONNECT-Headers auftauchen. Innerhalb von MIME-Nachrichten gelten die ursprünglichen Namen.

VER Vertreter, optional, mehrfach

Wenn eine persönliche Nachricht durch eine Vertreterin automatisch an eine andere Adresse weitergeleitet wird, so wird hier die Adresse der Empfängerin hinterlegt. Wird die Nachricht mehrfach durch Vertreterinnen weitergeleitet, so kann dieser Header auch mehrfach auftreten.

VIA Route-Information, optional, auch mehrfach, Reihenfolge beachten

Dieser Header war zur Drucklegung dieser Dokumentation noch nicht beschlossen. Bitte lesen Sie näheres in der Datei README auf der beiliegenden Diskette nach.

Durch diesen Header soll es möglich sein, stockende persönliche Nachrichten besser in den Griff zu bekommen. Die Reihenfolge dieses Header ist vorgeschrieben, und zwar fügt jedes weitere System, welches eine persönliche Nachricht transportiert, seinen VIA-Header hinter dem letzten vorhandenen VIA-Header ein. Die Behandlung ist unabhängig von Groß/Kleinschreibung. Für das Format gilt:

```
VIA: <Datum>@<system>.<Domain>
```

Das Datum wird dabei im Format wie bei EDA beschrieben angegeben.

Für die Uhrzeit gilt: Handelt es sich bei dem System um das erste System in der Kette, also dem Absender-System bzw. dem Server des Point-Systems, so kann die Uhrzeit in dieser einen VIA:-Line auf 0 Uhr gesetzt werden. (Datenschutz)

WAB Weiterleiten-Absender, optional, nur einmal

Die Absenderin der letzten Weiterleitung der Nachricht. Es gelten die für Adressheader gültigen Konventionen. (siehe Abschnitt III.5). Dieser Header wird beim Weiterleiten im Modul ‘Beibehalten des Absenders’ mit der E-Mail-Adresse des Weiterleitenden gefüllt.

ZUSAMMENFASSUNG optional, nur einmal

Kann eine kurze Zusammenfassung der Nachricht enthalten. Wenn die Zusammenfassung im Umfang 10% der Nachricht übersteigt, kann der Spooler eine Zustellung mit dem Verdacht auf Umgehung der Routmailgrenzen verweigern.

III.12 Weitere Headerzeilen

Neue Headerzeilen können jederzeit definiert werden. Jede Software muß ihr unbekannte Zeilen unverändert weitergeben. Für lokale Erweiterungen wird garantiert, daß niemals eine Headerzeile mit "X-" beginnen wird. Sie können also gefahrlos eigene Headerzeilen wie z.B. "X-Euromail-Version: 22.5" erzeugen.

Headerzeilen, die mit "U-" beginnen, sind UUCP-Header, entsprechen also RFC822 / 1036 bzw. entsprechenden Nachfolgestandards. UUCP oder Internet-Gateways können auf diese Weise Informationen, für die es im ZCONNECT zur Zeit noch keine Entsprechung gibt, 1:1 durchreichen. Beispiel:

```
U-Date:  Thu, 12 Jan 1987 PDST
```

entspricht der RFC-1036 Header-Zeile

```
Date:  Thu, 12 Jan 1987 PDST
```

In diesem Fall kann die Information jedoch gleichwertig mit dem EDA Header transportiert werden kann, es ist ja nur ein Beispiel...

Headerzeilen, die mit "F-" beginnen sind Header, die bei der Konversion von Nachrichten aus dem FTS0001-Format (und dessen Weiterentwicklungen) entstanden sind.

III.13 Verschlüsselung von Nachrichten mit PGP

Zum Verständnis dieses Dokumentes und vor allem zur sinnvollen Integration der hier beschriebenen Techniken in eine Point- oder MailBox-Software ist es unbedingt erforderlich, Phil Zimmerman's Anleitung (pgpdoc1.txt und pgpdoc2.txt) gelesen und verstanden zu haben!

Die Integration von PGP in eine MailBox unterliegt einem inhärenten Zielkonflikt. Alle UserInnen, die die hier vorgestellten Möglichkeiten zur Verbreitung ihres Public-Key verwenden, müssen sich bewußt sein, daß sie entweder ihrem lokalen Sysop oder dem Hersteller der MailBox-Software trauen. So wie es SystembetreiberInnen von ZERBERUS-MailBoxen unmöglich ist, den Inhalt der persönlichen Postfächer ihrer UserInnen einzusehen oder gar zu manipulieren, kann eine MailBox-Software auch die Unantastbarkeit der in der MailBox hinterlegten Public-Keys gewährleisten.

Es gibt (nicht standardisierte) Integrationen von PGP in andere Mail / News- Systeme. Die hier beschriebene Methode geht an vielen Stellen andere Wege, da ZCONNECT als neuer Standard noch nicht auf Tausende von Realisierungen Rücksicht nehmen muß und deshalb eine vollautomatische Integration möglich ist.

Wichtige Hinweise:

- Die in den folgenden Beispielen verwendeten Public Key's sind nicht und waren zu keinem Zeitpunkt gültige Public Keys.
- PGP und die im Anhang abgedruckten Beispielprogramme sind vom Autor in die Public Domain übergebene Programme. Sie können sie ohne jede Bedingung frei für jeden beliebigen Zweck verwenden.
- Die für die Beispiele verwendete PGP Version 2.3 enthielt schwerwiegende Fehler, die unter DOS z.B. zum Systemabsturz führen können. Bitte benutzen Sie eine aktuelle PGP Version.

III.13.1 Ziele der PGP Integration

Der große Vorteil eines Public-Key Verfahrens für den MailBox-Einsatz ist die freie Veröffentlichbarkeit der Public-Keys. Die große Gefahr ist die Manipulation von Public-Keys. Wir haben daher zwei Verfahren vorgesehen, diese Key's zu verbreiten:

- Der Public-Key wird mit anderen Nachrichten (öffentliche und nichtöffentliche) mitgeschickt. Diese Form der Verbeitung unterliegt verschiedenen Manipulationsmöglichkeiten. Keys, die über diesen Weg empfangen wurden, sollten anderweitig (z.B. per Telefon und Fingerprint) verifiziert werden.

  Programmierer von Pointprogrammen werden gebeten, die BenutzerInnen im Sinne einer effizienten Ausnutzung der Netzkapazität in ihrer Dokumentation zu einem sparsamen Umgang mit dieser Möglichkeit aufzurufen. Die Möglichkeit, den öffentlichen Schlüssel einer Gegenüber auf Anfrage automatisch zuzusenden, sollte eine einfache und breite Verteilung der öffentlichen Schlüssel gewährleisten.
- Der Public-Key wird in der Heimat-MailBox der Benutzerin hinterlegt und kann von dort mit Hilfe eines ZCONNECT-Anrufes (auch Gast-Anruf ohne Passwort) abgerufen werden. Die Verfügbarkeit eines Keys auf diesem Weg wird durch einen entsprechenden Status-Header in allen Nachrichten eines Users dokumentiert. Die Informationen in diesem Header genügen für einen Point, um automatisch einen Key-Request Anruf durchzuführen. Auf diesem Weg sind Public-Keys recht sicher abrufbar - allerdings ist immer noch Vertrauen in die Heimat-MailBox nötig. UserInnen, die selbst kein PGP verwenden laufen Gefahr, daß der Sysop der Heimat-MailBox einen Key für die UserIn generiert und als verfügbar markiert, obwohl die UserIn nichts davon weiß.

Im Zweifelsfall sollte immer ein Abgleich der Fingerprints vorgenommen werden.

Neben der Key-Verteilung sind die übrigen Ziele:

- Einfache Aufnahme der empfangenen Keys in den eigenen Key-Ring: dies kann vollautomatisch geschehen. Durch das "Introducer" Prinzip kann eine indirekte Authentisierung der Keys erfolgen. PGP verwaltet nicht-authorisierte Keys im Keyring selbst, so daß die UserIn sie später (z.B. telefonisch) verifizieren kann.
- Transparente Anzeige empfangener verschlüsselter oder unterzeichneter Nachrichten. Die UserIn soll keinen Unterschied in der Bedienung beim Lesen solcher Nachrichten feststellen können - lediglich einen Status- Hinweis "diese Nachricht ist echt" bzw. "diese Nachricht war verschlüsselt".

III.13.2 Key Repräsentation

PGP verwaltet Key's (sowohl öffentliche als auch geheime) vollständig selbstständig in einer geheimen und einer öffentlichen Datenbank, sogenannten "Key Rings". Key's werden aus dieser Datenbank mit dem Kommando `pgp -kx` in eine externe Datei kopiert. Das Ergebnis ist eine Binärdatei. Um das Verschicken von Key's auch in E-Mail Systemen ohne Binärnachrichten zu ermöglichen, wird die verpackte ("armor") Form benutzt, diese erhält man mit dem Kommando "`pgp -kxa`".

Mit "Key" ist im folgenden immer die rein binäre Form gemeint, eine verpackte Kodierung ist für `ZCONNECT` nicht erforderlich, Key's können als Binärnachricht verschickt werden bzw. werden in der `ZCONNECT` Online-Phase als Binärdatei übertragen.

Die verpackte Form des Headers entspricht der Base64 Kodierung gemäß RFC 1113[3], ergänzt mit einer CRC Prüfsumme. Base64 unterteilt die Binärdatei in 3-Byte-Gruppen und erzeugt daraus 4 6-bit Sextets, die als jeweils 1 Zeichen kodiert werden. Das Alphabet für diese Kodierung (siehe unten) ist so gewählt, daß es den Transport in allen

[3] Siehe auch auf Diskette beigefügten Mime-Standard (RFC 1341), dort ist die Base64-Codierung ebenfalls erklärt

gängigen E-Mail Zeichensätzen übersteht.

RFC 1113 definiert zusätzlich noch zwei Zeichen: '=' dient als Füllzeichen und ist beim Dekodieren zu überlesen, '*' ist ein Gruppenseparator und tritt in den hier relevanten Anwendungen nicht auf. Zeilen werden (bis auf die letzte natürlich) nach genau 64 Zeichen umgebrochen.

Ein Beispiel für einen verpackten PGP-Key:

```
- -----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.3

mQCNAizSdXcAAAEEAKtB+PaAAnCDLdFGbp0K2f+OTw8ZdoqLqyZZeFz0KHBTIdxj
geNlGYpRoxRuEE1OaQVu4jnE0tslOlz2QExEtw8qpAEOfI6EupLamtLa9aXVgZ67
+Nn6L55mUZlhfEAuTjSoMBetOLfPb6ojfLrQGT7ZkRBzfQWmajHmupmZcUdhAAUR
tCdNYXJ0aW4gSHVzZW1hbm4gPG1hcnRpbkBiaS1saW5rLm93bC5kZT4=
=yo/b
- -----END PGP PUBLIC KEY BLOCK-----
```

Die inneren Zeilen dieses Blocks enthalten in den ersten vier Zeilen den Key. Dekodiert man die Base64 Kodierung, erhält man exakt den gleichen Key, wie ihn PGP mit 'pgp -kx' ausgibt. In der fünften Zeile (beginnend mit einem Füll-'=') befindet sich eine 24 Bit CRC Prüfsumme der Binärrepräsentation des Keys.

Soll der obige Key in einem `ZCONNECT` Header transportiert werden, wird er in einer Header-Zeile als Base64 kodiert, entspricht also genau den ersten vier Zeilen des obigen Beispiels, nur ohne Zeilenumbrüche:

```
PGP-PUBLIC-KEY: mQCNAizSdXcAAA....saW5rLm93bC5kZT4=
```

Das Programm `MKKEY.C` erzeugt einen solchen Header aus der Binärrepäsentation des Keys. Dieses Programm ist im Anhang abgedruckt.

Die nötigen Routinen zur Dekodierung eine solchen Headers befinden sich in der Datei `armor.c` in den PGP-Sourcen, sind aber auch straightforward selbst entwickelbar. Damit ist es z.B. einem RFC Gateway möglich, einen `PGP-PUBLIC-KEY`-Header in einen an die Nachricht angehängten Public-Key Block zu verwandeln.

III.13.3 Unterschriften

Bei Nachrichten, die nur unterschrieben aber nicht verschlüsselt sind, steht die Unterschrift in einem Header namens `PGP-SIG`. `PGP` unterstützt das mit den Optionen `-sb`, die eine vom Dokument getrennte Unterschrift erzeugen. Mit dieser Unterschriftsdatei wird dann analog zur Binärrepräsentation der Schlüssel verfahren. Dies ermöglicht auch das Unterschreiben von Binärdateien.

Bei öffentlichen Nachrichten sollte diese Methode grundsätzlich benutzt werden, damit auch BenutzerInnen, die kein PGP installiert haben, die Nachricht lesen können.

Es ist wenig sinnvoll, bei jeder derartigen Nachricht den eigenen Public-Key (mit dem die Unterschrift verifiziert werden kann), in einem `PGP-PUBLIC-KEY` mitzuschicken. Wenn unterwegs die unterschriebene Nachricht gefälscht wird, ist es auch kein Problem, den anhängenden Schlüssel gleich mitzufälschen.

III.13.4 ZCONNECT Key Requests

In der ZCONNECT Online-Phase kann (analog zum `FILEREQ` Header) mit dem Header `PGP-KEYREQ` der Key einer UserIn abgefragt werden. Als Argument wird eine Adresse in ZCONNECT Notation angegeben. Falls vorhanden wird der Key der angegebenen UserIn als Datei übertragen.

Die UserIn muß dabei nicht unbedingt eine lokale Adresse auf der angerufenen MailBox haben, "vertrauenswürdige Server" können den Key-Request auch für UserInnen fremder MailBoxen anbieten.

III.13.5 Durch PGP geänderte Headerzeilen

Nach dem Verschlüsseln einer Nachricht kann es vorkommen, daß die Inhalte alter Header nicht mehr stimmen, nach dem Entschlüsseln aber wieder benötigt werden, zum Beispiel TYP:. Der Inhalt solcher (auch zukünftiger) Header wird daher grundsätzlich in einen Header mit einem vorangestellten ‘CRYPT-CONTENT-’ übernommen. ‘CRYPT’ deshalb, weil dieses Problem grundsätzlich auch für andere Verschlüsselungsverfahren gilt und die Benennung somit vereinfacht werden kann.

Anhang A: Empfehlung für den “MAPS-Roboter”

In der Vergangenheit sind sehr sehr viele, oft zueinander inkompatible Verfahrensweisen für automatische Brettbestellungen in verschiedene Software-Systeme eingebettet worden. Das Ergebnis war, daß Pointprogramme oft 5 oder mehr verschiedene Standards für Brettbestellungen implementieren mußten. Um mit diesem Chaos aufzuräumen, wurde auf dem Z-Netz-Treffen, Hamburg’94 ein Kompromiß erdacht, der in diesem Dokument als Anhang strengstens zur Implementierung empfohlen werden soll. Der hier beschriebene Standard ist so weit wie möglich zu den bisher geltenden Standards kompatibel und stellt insofern ein Minimum dar, das die verschiedenen ZCONNECT-Systeme zur Verfügung stellen sollten.

A.1 Allgemeines

Die Steuernachrichten sind an den Usernamen “MAPS@<system>.<domain>” zu schicken. Die Anweisungen stehen grundsätzlich im Betreff und ermöglichen somit eine Erstellung von Anweisungen “von Hand”. Benötigt ein Befehl weitere Parameter, so bilden diese den Nachrichtentext. Ist ein Befehl unbekannt, so sendet das System den Hilfstext mit dem Betreff “Your Help” an die Anfragende zurück. Dieser Hilfetext sollte eine Liste sämtlicher Befehle und Anweisungen zur Extraktion weiterer Hilfen enthalten.

Alle Befehle sind immer case insensitiv, so daß gilt “Help” = “help” = “hELP” = ...

Die Antworten des Programms sollten den Betreff: “Your ¡Befehl¿” erhalten und zur genaueren Referenzierung des Befehls mit dem BEZ-Header ausgestattet werden.

A.2 Standardbefehle

Dieser Abschnitt beschreibt die Befehle, die eine MAPS-Implementation mindestens akzeptieren muß, um sich ZCONNECT-MAPS-konform zu nennen. Dies sind die Befehle HELP zur Anforderung eines Hilfetextes, der Befehl LIST zur Anforderung einer Brettliste sowie die Befehle ADD und DEL zum Eintragen und Austragen von Brettern.

A.2.1 HELP

Dieser Befehl veranlaßt das System einen Hilfstext an die Anfragende zu versenden. Der Betreff der Antwort ist “Your HELP”. Im versendeten Hilfetext sollten alle verfügbaren Befehle aufgelistet sein. Es sollte außerdem beschrieben werden, wie u.U. weitere Hilfedateien angefordert werden können.

A.2.2 LIST

Dieser Befehl veranlaßt das System eine komplette Liste der verfügbaren Bretter an den Anfrager zurückzusenden.

Als Antwort wird eine Liste im fixen (!) Format übergeben. Der Betreff der Antwort lautet “Your LIST”. Die Länge einer Zeile ist nicht begrenzt!

Das Format definiert sich wie folgt:

Pos. 1	Steuerzeichen
Pos. 2	Leerzeichen, Ascii 32.
Pos. 3ff	Brettname. Nach dem Brettnamen kann, vom Namen durch Leerzeichen abgetrennt, optional noch eine Brettbeschreibung folgen.

Folgende Steuerzeichen sind definiert:

- '+' Brett ist derzeit bestellt.
- ' ' Brett ist derzeit nicht bestellt, aber bestellbar.
- '-' Brett ist nicht bestellbar.
- '!' Brett ist bestellt, kann aber nicht abbestellt werden (Zwangsanschluß).
- ';' Zeile enthält einen Kommentar.

Unbekannte Steuerzeichen sollten wie eine Kommentarzeile behandelt werden.

Der Brettname beginnt grundsätzlich mit einem Slash '/'. Unterbretter sind ebenfalls mit einem Slash gekennzeichnet. Er wird komplett in Großschreibung geliefert.

Für die Beschreibung wird keine maximale Länge festgelegt. Der für die Beschreibung gewählte Zeichensatz kann im Header CHARSET angegeben werden.

A.2.3 ADD

Dieser Befehl veranlaßt das System die angeforderten Bretter einzutragen, soweit zulässig.

Der Nachrichtentext enthält zeilenweise, beginnend am Zeilenanfang die Brettnamen ohne Beschreibung. Die Namen werden case insensetiv behandelt.

Ist das erste Zeichen einer Zeile kein Slash ('/'), so ist die Zeile nicht zu berücksichtigen.

Die Antwort enthält ein Protokoll, wobei das Format der Antwort von dem Befehl LIST entspricht. Protokolliert werden jedoch nur die angeforderten Bretter. Der Betreff der Antwort lautet "Your ADD".

A.2.4 DEL

Dieser Befehl veranlaßt das System die angegebenen Bretter auszutragen, soweit zulässig.

Der Nachrichtentext enthält zeilenweise, beginnend am Zeilenanfang die Brettnamen ohne Beschreibung. Die Namen werden case insensetiv behandelt.

Ist das erste Zeichen einer Zeile kein Slash ('/'), so ist die Zeile nicht zu berücksichtigen.

Die Antwort enthält ein Protokoll, wobei das Format der Antwort von dem Befehl LIST entspricht. Protokolliert werden jedoch nur die angegebenen Bretter. Der Betreff der Antwort lautet "Your DEL".

A.3 Erweiterte Befehle

Dieser Abschnitt beschreibt erweiterte Befehle, die die Arbeit eines Point-Programmes vereinfachen. Werden diese Befehle alle unterstützt, so kann sich eine MAPS-Implementation ZCONNECT-MAPS-kompatibel nennen.

A.3.1 HOLD

Dieser Befehl ist die sogenannte Urlaubsfunktion. HOLD ON veranlaßt das System an die Anfragende solange keine öffentlichen Nachrichten mehr zu schicken, bis diese den Befehl HOLD OFF gesendet hat. Diese Funktion eignet sich auch bei Systemen, die eine gewisse Zeit nicht mehr angerufen haben (Plattencrash, vorübergehend offline).

Die Antwort hat den Betreff "Your HOLD ON" bzw. "Your HOLD OFF". Sie enthält keinen Nachrichtentext.

A.3.2 INDEX

Dieser Befehl veranlaßt das System, Daten über die in bestimmten Brettern vorhandenen Nachrichten zurückzugeben.

Diese Funktion sollte auch für system-fremde Anwenderinnen verfügbar sein (Archiv-Systeme).

Der Nachrichtentext enthält zeilenweise, beginnend am Zeilenanfang die Brettnamen ohne Beschreibung. Die Namen werden unabhängig von Groß/Kleinschreibung behandelt. Zeilen die nicht mit einem Brettnamen beginnen (/ in der ersten Textposition) werden wie Kommentarzeilen behandelt.

Ist das erste Zeichen einer Zeile kein Slash ('/'), so ist die Zeile nicht zu berücksichtigen.

Der Betreff der Antwort lautet "Your INDEX". Sie enthält die ZConnect-Header der entsprechenden Nachrichten, wobei gilt

- Eine Leerzeile definiert das Ende eines Headers
- LEN enthält die tatsächliche Größe der Nachricht
- Es wird nur derjenige EMP mitgeliefert, der aufgrund der Anfrage relevant ist. Alle anderen werden entfernt.
- Alle mit F-, G-, U-, X-, Z- und ZNETZ- beginnenden Header sowie der ROT-Header, der GATE-Header sowie der MAILER-Header können, müssen aber nicht durch die Implementation gelöscht werden.

A.3.3 ORDER

Mit diesem Befehl können gezielt Nachrichten bestellt werden, soweit vorhanden.

Diese Funktion sollte auch für system-fremde Anwenderinnen verfügbar sein (Archiv-Systeme).

Die Parameter im Nachrichtentext haben folgendes Format:

```
<Brett><White Space><Message-Id>CrLf
```

Beispiel:

```
/Z-NETZ/!WICHTIG            fgsweeddssd.24@ldb.han.de
/T-NETZ/ZCONNECT/DISKUSSION 12345@bionic.zer.de
```

Der Brettname ist case-insensetiv und muß mit dem Slash ('/') beginnen. Die Message-Id ist bis zum "@" case-sensitiv, dahinter case-insensitiv.

Die Antwortnachricht enthält als Betreff "Your ORDER" und enthält einen ZConnect-Puffer, in dem diejenigen Nachrichten enthalten sind, die bestellt wurden und geliefert werden konnten.

Ein Protokoll (z.B. mit Kostenabrechnung etc.) kann dieser Nachricht als Kommentar (Header KOM) vorangestellt werden.

Es können abhängig von Route-Limits mehrere Nachrichten erstellt werden. Dabei darf aber nicht eine Nachricht in mehrere Teilnachrichten aufgeteilt werden.

A.4 Anmerkungen

Im Gegensatz zu der in den Wahlen verabschiedeten Version wird hier keine Implementation der Befehle ORDER-PM und FILES gefordert. Die Gründe dafür sind: ORDER-PM unterscheidet sich nur durch verstümmelte Header von ORDER, FILES dient zur Implementation eines Fileservers. Der Fileserver kann aber wesentlich effizienter durch den in Kapitel II beschriebenen ZCONNECT-Filerequest implementiert werden.

Index