

Netz – Beschreibung

Stand: 08.08.88

Alle Rechte vorbehalten.

Insbesondere ist die Überführung in maschinenlesbare Form, sowie das Speichern in Informationssystemen, auch auszugsweise, nur mit schriftlicher Genehmigung der GMD gestattet.

Herausgeber:

Gesellschaft für Mathematik und Datenverarbeitung mbH

Postfach 1240, Schloß Birlinghoven

D – 5205 Sankt Augustin 1

Telefon(02241) 14 – 1, Telex 8 89 469 gmd d

Telefax(02241) 14 28 89, BTX *43900#

Teletex 2627 – 224135 = GMDVV

Autor:

Dietmar Heinrichs

Texterstellung:

Dieser Text wurde mit der EUMEL – Textverarbeitung erstellt und aufbereitet und mit dem Agfa Laserdrucksystem P400 gedruckt.

Umschlaggestaltung:

Hannelotte Wecken

Hinweis:

Diese Dokumentation wurde mit größtmöglicher Sorgfalt erstellt. Dennoch wird für die Korrektheit und Vollständigkeit der gemachten Angaben keine Gewähr übernommen. Bei vermuteten Fehlern der Software oder der Dokumentation bitten wir um baldige Meldung, damit eine Korrektur möglichst rasch erfolgen kann. Anregungen und Kritik sind jederzeit willkommen.

Inhaltsverzeichnis

1. Einleitung	2
Teil 1: Netz einrichten und benutzen	
1. Benutzung des Netzes	3
2. Hardwarevoraussetzungen	6
3. Einrichten des Netzes	6
4. Informationsmöglichkeiten	8
5. Eingriffsmöglichkeiten	9
6. Fehlerbehebung im Netz	9
Teil 2: Arbeitsweise der Netzsoftware	
1. Die Netztask	12
2. Protokollebenen	15
3. Stand der Netzsoftware	20

1. Einleitung

Das EUMEL – Netz dient dazu mehrere EUMEL – Rechner (sog. Stationen) miteinander zu koppeln. Diese Kopplung wird von Betriebssystem dazu benutzt, das Sendungskonzept (siehe Systemhandbuch, Intertaskkommunikation) so auszudehnen, daß Tasks verschiedener Stationen einander Datenräume zusenden können. Auf dem Sendungskonzept aufbauende Konzepte nutzen daher automatisch das Netz aus: So ist es z.B. möglich

- von einer Station aus auf einer anderen zu Drucken,
- in PUBLIC einer anderen Station Dateien zu sichern (save), vorausgesetzt, daß PUBLIC dort ein free global manager ist,
- auf einer anderen Station zu archivieren (z.B. wenn das eigene Archivlaufwerk defekt ist oder ein anderes Format hat).

Das Netz kann ab EUMEL – Version 1.7.3 eingesetzt werden.

Teil 1: Netz einrichten und benutzen

1. Benutzung des Netzes

Zur Benutzung des Netzes stehen folgende Operatoren und Prozeduren zur Verfügung:

1.1

TASK OP / (INT CONST station, TEXT CONST taskname)

liefert die Task **taskname** von der Station **station**.

Wenn die Station **station** nicht am Netz ist oder nicht eingeschaltet ist, wird solange gewartet, bis das der Fall ist.

Fehlerfälle:

- task "..." gibt es nicht

Die angeforderte Task gibt es in der Zielstation nicht.

- Collectortask fehlt

Das Kommando **define collector** wurde nicht gegeben (siehe 4.2).

- Station x antwortet nicht

Eine nicht vorhandene oder abgeschaltete Station wurde angesprochen.

Hinweis: Dieser Fehler wird angenommen, wenn eine Überwachungszeit von ca. 30 Sekunden verstrichen ist, ohne daß Station x die Taskidentifikation angeliefert hat.

Beispiel:

list (5/"PUBLIC")

Dateiliste von PUBLIC auf Station 5 wird angefordert.

1.2

TASK OP / (INT CONST station, TASK CONST task)

liefert

station / name (task) .

Beispiel:

list (4/archive)

1.3

INT PROC station (TASK CONST task)

liefert die Stationsnummer der Task **task**.

Beispiel:

put (station (myself))

gibt die eigene Stationsnummer aus.

1.4

PROC archive (TEXT CONST archivename, INT CONST station)

dient dazu das Archiv auf der Station **station** anzumelden.

Beispiel:

archive ("std", 4); list (4/archive)

gibt das Inhaltsverzeichnis der Archivfloppy im Laufwerk der Station 4 aus.

Hinweis: Vergessen Sie bei solchen Querarchivierungen nicht die Stationsangabe bei jedem einzelnen Archivkommando (z.B. fetch ("xxx", 4/ archive).

Hinweis: Querarchivieren ist langsam. Verwenden Sie es nur, wenn Sie Floppyformate umsetzen wollen.

1.5

PROC free global manager

dient dazu, die eigene Task über das Netz ansprechbar zu machen. Jede andere Task im Netz kann dann die üblichen Manageraufrufe ('save', 'fetch', u.s.w.) an die eigene Task machen, sofern diese nicht an ein Terminal gekoppelt ist.

Die Task wird (wie bei 'break') abgekoppelt und meldet sich in Zukunft mit 'maintenance' statt mit 'gib kommando'.

Beispiel:

An Station 4 ruft man in der Task 'hugo' das Kommando **free global manager** auf. Anschließend kann man von jeder Station aus z.B. 'list (4/"hugo")' u.s.w. machen.

1.6

TEXT PROC name (TASK CONST t)

Diese (schon immer vorhandene) Prozedur wurde dahingehend erweitert, daß der Name einer Task einer anderen Station über Netz angefordert wird.

Fehlerfall:

Station x antwortet nicht

2. Hardwarevoraussetzungen

2.1 Zwei Stationen

Sie können zwei Stationen miteinander vernetzen, wenn Sie dafür an jeder Station eine V24 – Schnittstelle zur Verfügung stellen.

Diese beiden Schnittstellen verbinden Sie mit einem Kabel zur Rechnerkopplung (siehe Systemhandbuch, Teil 2).

2.2 Mehrere Stationen

Wenn Sie mehr als zwei Stationen vernetzen wollen, brauchen neben je einer V24 an jeder Station noch je eine Netzanschlußbox.

Jede Box besitzt eine V24 – Schnittstelle zum Anschluß an die V24 – Schnittstelle der zugeordneten Station und eine weitere Schnittstelle zur Verbindung der Boxen untereinander.

3. Einrichten des Netzes

Hinweis: Dieses Kapitel ist nur für Systembetreuer wichtig.

3.1 Legen Sie Stationsnummern für die am Netz beteiligten Rechner fest (von 1 an aufsteigend).

Die Boxen haben ebenfalls Stationsnummern. Die Stationsnummern der Box und des zugeordneten Rechners müssen übereinstimmen.

3.2 Holen Sie an jeder Station die Task **configurator** an ein Terminal und geben Sie das Kommando **define collector ("net port")**. Geben Sie außerdem das Kommando **define station (x)**, wobei x die gewählte Stationsnummer ist.

Hinweis: Taskkommunikationen, die zu dem Zeitpunkt laufen, führen zu fehlerhaftem Verhalten. Dies liegt daran, daß durch **define station** alle Task – Id's geändert werden müssen, weil eine Task – Id u.a. die Stationsnummer der eigenen Station enthält (siehe 2.3). TASK – Variable, die noch Task – Id's mit keiner oder falscher Stationsnum-

mer enthalten, können nicht mehr zum Ansprechen einer Task verwendet werden.

Beispiel: Der Spoolmanager (siehe Systemhandbuch, Teil 8) richtet beim Kommando **start** einen Worker ein und merkt sich dessen Task – Id in einer TASK – Variablen, um sicherzustellen, daß nur der Worker Dateien zum Drucken abholt. Wird jetzt das Kommando **define station** gegeben, kann der Spoolmanager seinen Worker nicht mehr identifizieren, weil der Worker eine neue Task – Id erhalten hat. Man muß daher den Worker löschen und mit dem Kommando **start** im Spoolmanager wieder neu einrichten.

Sinnvollerweise gibt man **define station** sofort nach den Laden eines frischen Systems von Archiv.

Konfigurieren Sie mit dem Kommando **configure** den für das Netz vorgesehenen Kanal auf

- transparent
- 9600 Baud (Standardeinstellung der Boxen)
- RTS/CTS – Protokoll
- großen Puffer
- 8 bit
- even parity
- 1 stopbit.

Falls diese Einstellungen nicht alle angeboten werden, klären Sie mit Ihrem Rechnerlieferanten, ob und wie diese Einstellungen erreicht werden können.

Hinweis: Notfalls kann auf das RTS/CTS – Protokoll verzichtet werden, wenn der Eingabepuffer der Station groß genug ist. Die Anzahl simultan laufender Netzkommunikationen ist dann auf

puffergröße DIV 150

begrenzt (bei Z80, 8086: 3; bei M20: 10).

Hinweis: Es können auch andere Baudraten (2400, 4800, 19200) an der Box eingestellt werden.

3.3 Achten Sie bei der Verbindung von der Station zur Netzbox (bzw. zur Gegenstation bei einem Zweistationennetz ohne Boxen) darauf, daß neben den Empfangs – und Sendeleitungen auch die Leitungen RTS und CTS verdrahtet

werden, also ein 5 poliges Kabel verwendet wird (siehe Systemhandbuch, Teil 2). Die Pin – Belegung der Boxen entspricht den dortigen Angaben.

Beispiel:

Verbindung eines CSK – Systems mit der Box:

Stecker Pin		Stecker Pin
2	<----->	3
3	<----->	2
4	<----->	5
5	<----->	4
7	<----->	7

3.4 Richten Sie eine Task **net** unter **SYSUR** ein und insertieren Sie dort die Dateien

```
net report/M
basic net
net manager/M.
```

Beantworten Sie die Frage nach dem Kanal für das Netz und nach der Flußkontrolle (RTS/CTS).

4. Informationsmöglichkeiten

In der Task **net** wird eine Datei **report** geführt in der Fehlersituationen des Netzes verzeichnet werden. Diese Datei kann in jeder anderen Task mit **list (/net)** angezeigt werden.

In jeder Task kann durch das Kommando **list (/net port)** eine Übersicht über die momentan laufenden Netzübertragungen der eigenen Station erhalten werden.

5. Eingriffsmöglichkeiten

5.1 Jede Task kann Sende- und Empfangsströme, die bei **list (/net port)** gemeldet worden sind und die eigene Task betreffen, abbrechen. Hierzu ist das Kommando **erase ("x"/net port)** zu geben, wobei x die Stromnummer (aus dem 'list') ist.

Unberechtigte Löschversuche werden abgewiesen.

Von der Task 'net' aus können jedoch damit beliebige Ströme abgebrochen werden.

5.2 Durch das Kommando **start** kann von der Task 'net' aus das Netz neu gestartet werden. Dabei werden alle augenblicklichen Netzkommunikationen gelöscht. Die Tasks 'net port' und 'net timer' werden dabei gelöscht und neu eingerichtet.

start (kanal, quit) wirkt wie **start**. Zusätzlich wird als Netzkanal 'kanal' eingestellt und maximal 'quit' Empfangsströme zugelassen. 'quit' ist auf 3 zu setzen, wenn der Kanal ohne RTS/CTS angeschlossen ist (siehe 3.2).

6. Fehlersuche im Netz

Fehler im Netz können sich verschiedenartig auswirken. Im Folgenden wird auf einige Beispiele eingegangen:

Beispiel:

Auf **list (4/public)** erfolgt die Meldung 'Station 4 antwortet nicht'.

Fehlermöglichkeiten:

- Station 4 gibt es nicht am Netz.
Abhilfe: Richtige Station angeben.
- Station 4 ist nicht eingeschaltet.
Abhilfe: Station 4 einschalten. Kommando erneut geben.
- Netztask an Station 4 ist nicht arbeitsfähig.
Abhilfe: Kommando 'start' in der Task 'net'.

- Stationsnummern und Boxnummern stimmen nicht überein.
Abhilfe: Mit 'define station' Stationsnummern korrigieren (siehe 3.2).
- Verbindung Rechner/Box am eigenen Rechner oder an Station 4 fehlt.
Abhilfe: Verbindungen überprüfen. Durch Ansprechen einer dritten Station kann oft schnell geklärt werden, welche Rechner/Box – Verbindung defekt sein muß.
- Verbindung der Boxen untereinander defekt.
Abhilfe: Fehlende Verbindung, Masseschluß und Dreher (keine 1:1 Verbindung) überprüfen und beheben.
Hinweis: Liegt z.B. ein Masseschluß vor, so kann es durchaus sein, daß Boxen, die nicht in der Nähe des Masseschluß stehen noch miteinander arbeiten können. Man kann aus der Tatsache, daß zwei Boxen miteinander arbeiten können, also nicht schließen, daß man nicht nach diesem Fehler suchen muß.

Beispiel:

Auf **list (4/public)** erfolgt keine Reaktion.

- Station 4 ist während dieser Sendung zusammengebrochen.
Abhilfe: Station 4 wieder starten. Die Bearbeitung des 'list' – Kommandos wird automatisch wieder aufgenommen.
 - PUBLIC auf Station 4 ist nicht im Managerzustand.
Abhilfe: PUBLIC in den Managerzustand versetzen.
- Hinweis: Das Netz hat noch nicht die volle Sendungslogik des EUMEL. So wird nur ca. 10 Minuten lang versucht, eine Sendung zuzustellen. Danach wird die Sendung gelöscht. Ist dies eingetreten, so muß das list – Kommando erneut gegeben werden.
- Fehler in der Netzhardware.
Überprüfen Sie, ob
 - die Boxen eingeschaltet sind,
 - die Bereitlampe blinkt (wenn nicht: RESET an der Box)
 - die V24 – Kabel richtig stecken,

- die Boxen untereinander verbunden sind (1 zu 1 Verbindungen der 5 poligen Diodenbuchsen).
- Die Netzsoftware ist auf einen nicht vorhergesehenen Fehler gelaufen. Dieser wird im Report vermerkt.
Abhilfe: Geben Sie in der Task **net** das Kommando **start**. Dadurch wird die Netzsoftware neu gestartet. Alle Netzkommunikationen dieser Station gehen verloren.

Beispiel:

Auf list (**4/public**) erfolgt die Meldung 'Collectortask fehlt'.

- In der Task 'configurator' wurde das Kommando 'define collector' (siehe 3.2) nicht gegeben.
- Die Task 'net port' existiert nicht mehr.
Abhilfe: Kommando 'start' in der Task 'net'.

Beispiel:

Nach **fetch ("hugo",4/public)** sind Teile von der Datei "hugo" verfälscht.

- Die V24 – Verbindung zur Box ist nicht in Ordnung.
Abhilfe: Abstand zwischen Rechner und Box verkürzen; Baudrate erniedrigen; Durch Wechseln der V24 – Schnittstelle feststellen, ob diese defekt ist.
Hinweis: Die Verbindung zwischen den Boxen ist durch Prüfsummen abgesichert (Hardware).

Teil 2: **Arbeitsweise der Netzsoftware**

1. **Die Netztask**

In diesem Kapitel wird beschrieben, wie eine Netztask in das System eingebettet ist und welche Aufgaben sie hat. Unter Einhaltung dieser Konzepte kann die ausgelieferte Netztask so geändert werden, daß sie beliebige andere Netzhardware unterstützt. Z.Zt. ist die Netzsoftware noch nicht so gegliedert, daß nur eine hardwareabhängige Komponente ausgetauscht werden muß.

Die Kommunikation zwischen Tasks im EUMEL – Betriebssystem basiert auf einem Rendezvouskonzept: Die Zieltask einer Sendung muß empfangsbereit sein, wenn die Quelltask sendet.

Die Kommunikationsprozeduren auf der niedrigsten Ebene sind 'send' (Senden) und 'wait' (Warten auf Empfang). Bei der Kommunikation werden eine Integer 'code' und ein Datenraum 'dr' übergeben. 'code' muß ≥ 0 sein, da negative Codes systemintern verwandt werden. Ist die empfangende Task an einen Kanal gekoppelt ('continue'), so führt eine Zeicheneingabe auf diesem Kanal dazu, daß eine Sendung mit dem Code -4 ankommt. Die Eingabedaten müssen mit den üblichen Eingabeprozeduren ('inchar' u.s.w.) abgeholt werden. Der übermittelte Datenraum und die Absendertask sind dabei ohne Bedeutung und dürfen nicht interpretiert werden.

Die Prozedur 'send' hat einen Rückmeldeparameter, der besagt, ob die Sendung übermittelt wurde. Gibt es die Zieltask nicht oder steht sie nicht im 'wait', so kann die Sendung nicht übermittelt werden.

Ein Entwicklungskriterium für das EUMEL – Netz war es, möglichst wenig Unterstützung von der virtuellen EUMEL – Maschine (EUMEL0) zu fordern, damit weitgehend in ELAN programmiert werden kann. Dadurch ist es möglich eine (privilegierte) Task mit der Netzabwicklung zu betrauen.

Zunächst wird auf die EUMEL0 – Unterstützung eingegangen:

1.1. Es gibt die Prozedur 'define collector', mit der die für das Netz verantwortliche

Task der EUMEL0 – Maschine bekannt gemacht wird. Diese Task wird im folgenden Collector genannt.

- 1.2. Es gibt die Prozedur 'define station', die für den Rechner eine Stationsnummer einstellt. Anhand dieser Nummer werden die Rechner eines Netzes unterschieden. Das Einstellen bewirkt, daß für alle Tasks die Stationsnummer in ihre Task – Id eingetragen wird (Task – Id's sind die Werte, die der Typ TASK annehmen kann).
- 1.3. Der Befehl 'station (task)' liefert die Stationsnummer der 'task'. So liefert z.B. 'station (myself)' die Stationsnummer des eigenen Rechners.
- 1.4. Eine Sendung, deren Zieltask in einem anderen Rechner liegt (also station (ziel) < > station (myself)), wird auf die Collectortask geleitet.
- 1.5. Es gibt eine Prozedur 'collected destination', die es dem Collector erlaubt, die eigentliche Zieltask einer auf ihn geleiteten Sendung zu erfahren.
- 1.6. Es gibt eine Variante der Prozedur 'send', die es dem Collector gestattet, der Zieltask eine beliebige andere Task als Absender vorzumachen.
- 1.7. Es gibt eine spezielle Task – Id 'collector', durch die der augenblicklich eingestellte Collector erreicht wird. Diese wird als Zieltask beim Aufruf der Vermittlungsdienste angegeben (siehe 2.5). Eine Sendung an 'collector' wird von EUMEL0 an den derzeitigen Collector geschickt.

Ein Collector kann also auf drei Wegen von den übrigen Tasks desselben Rechners Sendungen erhalten:

1. Über ein normales Send (z.B. bei 'list (/ "net port")', wenn "net port" der derzeitige Collector ist),
2. über ein Send an die Task 'collector' (s.u.) und
3. als umgeleitete Sendung (z.B. bei 'list' an eine Task auf einem anderen Rechner).

Der Collector kann diese Fälle anhand von 'collected destination' unterscheiden.

Die Punkte 1.4...1.6 dienen dazu, den Collector für über Netz kommunizierende Task unsichtbar zu machen: Der Collector taucht nicht als Ziel oder Quelle von Sendungen auf. Das ist notwendig, damit normale Tasks sich nicht darum kümmern müssen, ob

eine Sendung übers Netz geht oder im eigenen Rechner bleibt.

Wenn ein Datenraum an einen anderen Rechner geschickt wird, muß der gesamte Inhalt (z. Zt. max. 1 MB) übertragen werden. Dies macht bei der üblichen Netzhardware eine Zerlegung in Pakete nötig (siehe Systemhandbuch, Teil 4). Für Netze über V24 – Kanäle stehen spezielle Blockbefehle zur Verfügung:

1.8. blockin / blockout (dr,seite,512+abstand,anzahl,rest)

Es werden maximal 'anzahl' Bytes transferiert. In 'rest' wird zurückgemeldet, wieviel Bytes nicht bearbeitet wurden (z.B. weil der Kanal nichts anliefert). Bearbeitet werden die Bytes

'seite' * 512 + 'abstand'

bis maximal

'seite' * 512 + 'abstand' + 'anzahl' - 1

Der Kanal, an den die Task gekoppelt ist, wird dabei über Stream-IO (d.h. 'inchannel' bei 'blockin' bzw. 'out' bei 'blockout') angesprochen.

Hinweis: Die Anforderung darf nicht über Seitengrenze gehen, d.h.

'abstand' + 'anzahl' <= 512

muß erfüllt sein.

Eine Netzsendung läuft wie folgt ab:

Die Task q auf Rechner rq mache ein 'send' an die Task z auf Rechner rz.

1. Die Prozedur send ist ein EUMEL0 – Befehl. Die EUMEL0 – Ebene erkennt, daß die Sendung an die Station rz geht, da die Stationsnummer in der Task – Id enthalten ist. Daher wird die Sendung zum Collector, den EUMEL0 wegen 'define collector' kennt, umgeleitet.
2. Die Task Collector empfängt über 'wait' den Datenraum, den Sendecode und die Absendertask q. Die Zieltask z erfährt sie durch 'collected destination'.
3. Der Collector nimmt Kontakt mit dem Collector des Rechner rz, dessen Stations-

nummer ja 'station(z)' ist, auf und Übermittelt diesem Sendecode, Quelltask (q), eigentliche Zieltask (z) und den Datenraum. Da die Collectoren in ELAN geschrieben sind, können sie an beliebige Netzhardware und Protokolle angepaßt werden.

4. Der Collector auf Rechner rz verwendet das spezielle 'send', um der Zieltask die Sendung zuzustellen. Dadurch erscheint nicht der Collector sondern die Task q als Absender der Sendung.

Zur Abwicklung der Vermittlungsebene (Teil 1: 2.4) muß der Collector noch spezielle Funktionen beherrschen. Diese sind

der /-Operator (Taskname in Task-Id wandeln) und
die name-Prozedur (Task-Id in Namen wandeln).

Der /-Operator macht eine Sendung an den 'collector', wobei im Datenraum der Name der Task steht und der Sendecode gleich der Stationsnummer ist (siehe Quellcode, Packet tasks). Der Collector setzt sich mit dem Collector dieser Station in Verbindung, damit dieser die Task-Id ermittelt und zurückschickt. Der eigene Collector schickt dann dem /-Operator als Antwort einen Datenraum, der die Task-Id enthält.

Umgekehrt läuft 'name' ab: Wenn die Task-Id von einer fremden Station ist, schickt 'name' eine Sendung an den 'collector', wobei im Datenraum die Task-Id steht und Sendecode = 256 ist. Der Collector entnimmt die Stationnummer der Task aus der Task-Id und läßt sich vom entsprechenden Collector den Tasknamen geben. Dieser wird der 'name'-Prozedur im Antwortdatenraum übergeben.

2. Protokollebenen

In diesem Kapitel werden die Protokollebenen für das Netz beschrieben, wie sie die ausgelieferte Netzsoftware benutzt und erwartet. Bei anderer Netzhardware müssen die Ebenen 1 bis 3 ausgetauscht werden. Unter Einhaltung der im vorigen Kapitel beschriebenen Randbedingungen können auch die höheren Ebenen geändert werden.

2.1 Physikalische Ebene

2.1.1 Station < - - > Box

V24 – Schnittstelle mit RTS/CTS – Handshake. Volduplex.

2.1.2 Box < - - > Box

RS422 über 2 verdrehte Leitungspaare (Takt und Daten).

2.2 Verbindungsebene

2.2.1 Station < - - > Box

Asynchron

8 Bit

Even Parity

2400/4800/9600/19200 Baud (einstellbar über Lötbrücken)

2.2.2 Box < - - > Box

SDLC

400 KBaud

2.3 Netzebene

2.3.1 Station < - - > Box

Telegrammformat: STX, <n>, <ziel>, <quelle>, <(n-4) byte>

<n> ist Längenangabe ($8 \leq n \leq 160$)

<ziel>, <quelle> sind Stationsnummern. Diese müssen an den jeweiligen Boxen über Lötbrücken eingestellt sein.

Box - - > Station:

Ein Telegramm kommt nur bei der Station an, bei deren Box die Nummer <ziel> eingestellt ist. Dadurch ist ein Mithören fremder Übertragungen nicht möglich (Datenschutz).

Zwischen Telegrammen können Fehlermeldungen der Box (Klartext) übermittelt werden (z.B. 'skipped x', wenn ein STX von der Box erwartet wurde, aber 'x' von der Station ankommt).

Station - - > Box:

Ein Telegramm wird nur abgeschickt, wenn <quelle> mit der eingestellten Nummer übereinstimmt (Datenschutz: Man kann nicht eine belie-

bige Station zu sein vorschwindeln, es sei denn man hat physischen Zugriff zur Box und stellt dort die Stationsnummer um).

2.3.2 Box < -- > Box

Telegrammformat: FRAME, <ziel>, <quelle>, <daten> ,
<CRC-Code>

Eine Längenangabe ist nicht nötig, da SDLC eine Rekonstruktion der Länge erlaubt.

Telegramme mit falschen CRC-Code werden vernichtet. Auf höheren Ebenen muß dies durch Zeitüberwachung erkannt und behandelt werden.

2.4 Transportebene

Diese Ebene wickelt das Rendezvous zwischen einer Task, die 'send' macht, und einer Task, die im 'wait' steht, ab (siehe: EUMEL – Systemhandbuch).

Der im 'send' angegebene Datenraum wird als Folge von Seiten (im EUMEL – Sinne: Pagineinheit und Allokierereinheit) übermittelt, wobei jede Seite noch in 64 Byte große Stücke zerlegt wird. Es werden nur echt allokierte Seiten übermittelt. Um nicht jedes Telegramm voll qualifizieren zu müssen, wird zunächst eine Art virtuelle Verbindung durch ein OPEN – Telegramm eröffnet. Danach folgen variable viele DATA – Telegramme. Beide Sorten werden durch QUIT – Telegramme quittiert, um folgende Funktionen zu ermöglichen:

Flußkontrolle (z.B. Zielrechner langsam)
Wiederaufsetzen (verlorene Telegramme)
Abbruch (z.B. weil Zieltask inzwischen beendet).

Ein CLOSE – Telegramm ist nicht nötig, da das letzte DATA – Telegramm als solches erkannt werden kann (siehe unten).

2.4.1 OPEN – Telegramm

STX, 20, <ziel>, <quelle>, <strom>, <sequenz>, <seite>,
<quelltask>, <zieltask>, <code>

<ziel>, <quelle> siehe 2.3.1

- <strom> Die Stromnummer identifiziert die virtuelle Verbindung. Sie muß in den QUIT – Telegrammen angegeben werden.
- <sequenz> – 1 (Kennzeichen für OPEN)
- <seite> Nummer der ersten echt allokierten Seite des Datenraums (= – 1, falls Nilspace)
- <quelltask> Taskid der sendenden Task
- <zieltask> Taskid der empfangenden Task
- <code> Wert des im 'send' angegebenen Codes.

2.4.2 DATA – Telegramm

STX, 74, <ziel>, <quelle>, <sequenz>, <seite>, <64 byte>

<sequenz> wird von Telegramm zu Telegramm hochgezählt. Dient der Überwachung gegen verlorengegangene Telegramme bzw. durch Zeitüberwachung verdoppelter Telegramme.

<seite> Nummer der x.ten echt allokierten Seite des Datenraums. ($x = (\text{sequenz} + 16) \text{ DIV } 8$).

<64 byte> Nutzinformation. Diese gehört zur Adresse a des Datenraums.

$$a = N (\text{sequenz} \text{ DIV } 8 + 1) * 512 + (\text{sequenz} \text{ MOD } 8) * 64$$

wobei N (x) die Nummer der x.ten Seite ist.

Aus den Formeln ergibt sich, daß diese Nummer schon in einem vorhergehenden DATA/OPEN – Telegramm übermittelt wurde (im Feld <seite>).

2.4.3 QUIT – Telegramm

STX, 8, <ziel>, <quelle>, <strom>, <quit>

<strom> muß die Stromnummer sein, die in dem OPEN/DATA –

Telegramm stand, das quittiert wird.

- <quit>
- 0 : ok. Nächstes Telegramm schicken.
 - 1: Übertragung neu starten (mit OPEN), weil die Empfangsstation das OPEN nicht erhalten hat.
 - 2: Übertragung ca. 20 Telegramme zurücksetzen.
 - 3: Übertragung abbrechen.

2.5 Vermittlungsebene

Diese Ebene ist dafür zuständig, Tasknamen von Task auf anderen Stationen in Taskids (Werte des Typs TASK) zu wandeln und umgekehrt. Hierzu wird im entsprechenden OPEN-Telegramm der Code -6 (bzw. -7) als <code> eingetragen. Die Netzempfangstask erkennt diese Codes und wickelt die Aufgaben selbst ab, sodaß es dabei nicht nötig ist, irgendeine Taskid der Zielstation zu kennen.

Dieses Verfahren ist möglich, weil im 'send' nur positive Codes erlaubt sind.

2.6 Höhere Ebenen

Höhere Ebenen sind nicht mehr netzspezifisch. Sie basieren alle auf dem Send/Wait-Konzept des EUMEL. So gibt es z.B. den 'global manager', der Aufbewahrung und Zugriff von Dateien in einer Task regelt. Dabei darf diese Task (bei der Variante 'free global manager') auf einer beliebigen Station im Netz liegen. Wegen des Rendezvous-Konzepts können beliebige Sicherheitstrategien benutzt werden (z.B.: Keine Dateien an Station 11 ausliefern). Von großem Wert ist z.B., daß man ohne weiteres das Archiv (Floppylaufwerk) einer anderen Station anmelden und benutzen kann, wodurch eine einfache Konvertierung von Floppyformaten möglich ist. Dies ist möglich, weil auch die Archiv-Task der Stationen sich an das Globalmanagerprotokoll halten.

Bemerkungen

Fehlerbehandlung besteht bis Ebene 3 darin, fehlerhafte Telegramme einfach zu entfernen. Die Ebene 4 überwacht den Netzverkehr sowieso über Timeouts, die eine

Wiederholung eines Telegrammes bewirken, wenn die Quittung ausbleibt.

Da bei der sendenden Station der ganze Datenraum zur Verfügung steht, ist eine Fenstertechnik (wie bei HDLC) nicht nötig. Es kann zu jedem Zeitpunkt um beliebig viele Telegramme zurückgesetzt werden.

Da im EUMEL eine Textdatei ein Datenraum mit sehr komplexer Struktur ist (wegen der Insert/Delete – Möglichkeiten, ohne den Rest der Datei zu schieben), ist es ein hoher Aufwand, von einem fremden Betriebssystem her Textdateien ins EUMEL – Netz zu senden. Für solche Zwecke muß noch eine einfachere Dateistruktur definiert werden und entsprechende Dateikonverter erstellt werden.

3. Stand der Netzsoftware

Das EUMEL – System wickelt die Prozedur **send** über das Netz ab, wenn die Stationsnummer der Zieltask ungleich der eigenen Stationsnummer ist. Umgekehrt kann man der von der Prozedur **wait** gelieferten Absendertask die Absenderstation entnehmen (siehe Prozedur **station** in Abschnitt 3).

Nicht unterstützt wird z.Zt. die Logik der Prozeduren **call** und **pingpong**. Diese funktionieren nur in der gewohnten Weise, wenn die Zieltask in **wait** steht. Ist die Zieltask länger als ca. 10 Minuten **busy** oder nicht mehr vorhanden, geht die Sendung einfach verloren (Gefordert ist: bei **call**: immer wieder versuchen; bei **pingpong**: Rückmeldung – 2).

Wegen dieser Einschränkung kann man z.B. ein sicheres Drucken von Station a auf einen Drucker der Station b nur durch einen eigenen Spoolmanager auf Station a verwirklichen. Die Einrichtung eines solchen Managers ist allerdings sowieso sinnvoll, damit man

- das normale 'print' – Kommando verwenden kann
(statt z.B. save ("xxx", 4/printer);
und
- nicht zu warten braucht, bis die Datei übers Netz gesendet ist.