

# **TheNetNode**

**Made by**

# **Nord><Link**

**Ausgabe 16. November. 1998**

## **Benutzerhandbuch zu der Packet-Radio Knotensoftware**

### **TheNetNode (PC)/(GNU)/(Linux)/(TNC3) V 1.76**

#### **Änderungen gegenüber der Beschreibung von der Version TheNetNode 1.73**

##### **(Ausgabe zur Jahreshauptversammlung NORD><LINK 1996 in Hannover)**

- 1.) Die Files TNN170.xxx sind gegen TNN176.xxx ersetzt worden.
- 2.) ESC I 1... 6 ist entfallen. Die Einstellungen des Knoten Calls SSID usw. sind nur noch über die TNN176.PAS möglich. In der TNN176.TNB ist mit ESC I <CALL> <ALIAS> nun nur noch das HOST Call zu ändern. Nils DF6LN hat eine kleine EXE geschrieben, welche die TNN175.PAS nach TNN176.PAS umsetzt und entsprechend editiert. Damit sie nicht via FUNK neu und evtl. offen zum Knoten gesendet werden muß oder der Knoten aufgesucht werden muß. Programm heißt UPD\_PAS.EXE und ist mit im TNN Verzeichnis vorhanden.
- 3.) Die Datei TNN175.STA darf N I C H T nach TNN176.STA kopiert werden. Sie führt zu nicht löschbaren Einträgen. Knoten bitte O H N E diese Datei starten und mit s + <call> die Statistik neu einstellen.
- 4.) RUNBATCH <filename.tnb>. Hiermit lassen sich nun auch TNB direkt ausführen.
- 5.) Beim Interlink Setup wird nun auch der direkte Nachbar angegeben zu dem der Interlink Setup geroutet wird.
- 6.) Im Connect-Text läßt sich nun das aktuelle Datum des AKTUELL.TXT automatisch anhängen und anzeigen. Dazu ist im Connect-Text hinter dem Aktuell ein %A anzufügen.
- 7.) Version + zeigt nun an, mit welcher Zusammensetzung die TNN.EXE compiliert ist.
- 8.) Ab dieser Version ist das 30-Kanal Host-Mode Interface integriert das für den Anschluß einer Mailbox, Cluster oder auch eines Laptops dienen kann. Mit SET CONSOLE=<COM> muß eine Schnittstelle zugewiesen sein. (Befehle ähnlich TF27)
- 9.) Mitschnitte für die BISHERR ESC E1/E0 verwendet wurde muß nun ESC @E1/ @E0 benutzt werden.
- 10.) B und B+ für die MAILBOX ist im LINK Befehl entfallen. Mailbox wird nun mit L oder L+ eingetragen.
- 11.) Die Textmarkierungen in der TNN173.PAS wurden bisher mit einem # markiert. Sie wurden nun durch Semikolon ersetzt. Grund: Es war bisher nicht mehr möglich #ALIAS zu setzen.
- 12.) Es läßt sich nun unabhängig von einer MAILBOX am eigenen Knoten eine lokale Mailbox im der TNN176.TNB mit M <call> eintragen. M - löscht den Eintrag.
- 13.) Es läßt sich nun unabhängig von einem CLUSTER am eigenen Knoten ein lokales Cluster im der TNN176.TNB mit DX <call> eintragen. DX - löscht den Eintrag.
- 14.) Der größte Umbau allerdings ist: TNN arbeitet nun (außer zu alten Nachbarn) mit Laufzeiten. Die Umrechnung auf Qualitäten erfolgt immer dann, wenn kein Nachbar neuen Typs erkannt wurde.
- 15.) Das FlexNet-Interface wurde nun so weit verbessert, daß nicht nur die Destinations übernommen werden, sondern auch an andere FlexNet-Nachbarn weiter geleitet werden.
- 16.) SET TNNCFG = <path> ist entfallen.
- 17.) SET TNNCFG = <1.Anzahl>, <2.Anzahl>: 1.Anzahl ist die Größe der Nodes - / Destinations Liste 2.Anzahl ist die Länge der Linkliste (Parameter 1 entfallen).
- 18.) Einige <ESC> - Befehle lassen sich nun auch von Ferne eingeben. Z.B. <ESC> T wird von Ferne ausgeschrieben als „ESC T“ eingegeben. So muß er nun auch in der TNN176.TNB eingetragen werden.
- 19.) Unter Links ist nun auch N+ eintragbar, wenn bekannt ist, daß der Nachbar den neuen Router von TheNetNode unterstützt. N- ist nun für die Nachbarn die nur das NetRom - Protokoll unterstützen (alter Router).

- 20.) Das Ein-/Ausschalten von L2Digi ist entfallen. L2Digi ist nun grundsätzlich eingeschaltet.
- 21.) N < <Nachbar-Call> zeigt alle Nodes/Destinations die von diesem Nachbarn als „schnellster“ Weg gemeldet werden. N <Node-Call> \* zeigt ALLE Nachbarn an mit Ihren gemeldeten Zeiten.
- 22.) Ab der TNN176 werden nun nur noch Zeiten und KEINE Qualitäten mehr ausgetauscht.
- 23.) Neu ist auch der NRR (NetRom - Route - Rekord). Er wird mit N <CALL> ausgelöst. Es wird nun die Laufzeit zum Node angezeigt und ein Frame an das Node gesendet in dem der Weg dorthin und wieder zurück aufgezeichnet wird. NRR funktioniert natürlich nur bei Nodes, die NRR unterstützen und die NRR-Frames zurücksenden. Einige Digipeater leiten zwar die NRR-Frames weiter aber können es nicht beantworten. Sie stehen in der Wegliste dann mit einem „?“ .
- 24.) Änderung der QST-Bake. Sie sendet nun nur noch den eingestellten Bakentext. Die Linkmessungen werden nun an METRIC gesendet. Das vereinfacht das Protokollieren und Auswerten.
- 25.) Ausgabe um U <call> (mit und ohne SSID) erweitert. Ausgegeben werden nun die Einträge aus der U-Liste und U + Liste.
- 26.) STAT-Befehl geändert - man muß nun als Parameter angeben, welchen Teil der Statistik man angezeigt haben möchte. Die Statistik wurde um S E wie (S)tatistik (E)rrors, S H wie (S)tatistik (H)ardware, usw. erweitert. Die einzelnen Fehler sind nun je Kanal sichtbar. Mit S \* kann die ganze STAT abgerufen werden.
- 27.) Die U \* Liste zeigt nun wieder Byte an. Jedoch nur noch 6-stellig der Überlauf wird nun mit „k“ „M“ und wenn nötig mit „G“ gekennzeichnet.
- 28.) Die Port-Parameter (20 bis n) sind entfallen;  
L2-Digipeating ist nun generell eingeschaltet;  
TX-Port (senden auf einen anderen Port als dem RX-Port) entfallen;  
MTU und L2-RETRY ist intern festgelegt worden;  
PERSIST wird aus der Anzahl von Interlinks und Userlinks auf dem Port berechnet;  
SLOTTIME, T2-TIMER und IRTT wird auf Grund der Baudrate eingestellt.
- 29.) Die PARAMETER sind auf das nötige gekürzt worden.
- 30.) Neu dazugekommen ist eine graphische Ausgabe der Statistik. Siehe (G)raph.
- 31.) Die Port-Parameter sind nun nur noch über die Port-Befehle zu setzen und anzuzeigen (siehe PORT-Befehl).
- 32.) Das setzen von USERPORT/LINKPORT entfällt. Im Modebefehl wird nun ein Flag gesetzt ob die MH-Liste geführt werden soll.
- 33.) Ports die NUR Interlink Zwecken dienen können im Modebefehl auf SYSOP Status gesetzt werden.
- 34.) Neu geführt wird eine L3MH-Liste. Sie kann in ihrer Länge wie die bekannte MH-Liste gesetzt werden. L3MH = <anzahl>
- 35.) Eine Längenänderung der MH-Liste führt nicht mehr zum totalen Verlust aller Daten.
- 36.) Die Bedienung des PACSAT wurde etwas geändert.
- 37.) EDIT/LOAD umgearbeitet, Code eingespart und einige Bytes RAM durch Wegfall von editname / edittmp. EDIT/LOAD gehen jetzt über gemeinsame Variablen, gehen also nicht mehr gleichzeitig. Dafür geht es jetzt über alle Pfade hinweg (LOAD Z:\FOO\ABC.TXT).
- 38.) BayCom USCC werden nun direkt aus TNN unterstützt!
- 39.) Trace umgestaltet und erweitert.

## Benutzerhandbuch zu der Packet-Radio Knotensoftware

## TheNetNode (PC) V 1.76

Nachdruck mit Quellenangabe für nicht kommerzielle Zwecke erlaubt.

## Inhaltsverzeichnis

Änderungen gegenüber der Beschreibung von der Version TheNetNode 1.73 .....	2
Vorwort: .....	7
Das Programmpaket TheNetNode:.....	7
BAYCOM SCC-KARTEN: .....	7
DER TOKENRING .....	7
DER KISSLINK .....	8
DIE VANESSAKARTE.....	8
EXTERNE TREIBER .....	9
ETHERNET-SCHNITTSTELLE:.....	11
DPMI ODER NUTZUNG DES MEIST BRACHLIEGENDEN SPEICHERS ÜBER 1MB: .....	11
TheNetNode-Programm:.....	11
DAMA ... VERKEHRSREGELUNG DURCH DEN KNOTEN: .....	12
DAMA-Ablaufsteuerung:.....	12
SYSOP-ARBEITEN AM KNOTEN: .....	13
RECHNERKONFIGURATION: (HIER ALS BEISPIEL FÜR EINEN 386 ODER 486 MIT HD) .....	13
EMM386 BZW. QEMM ZUR BEREITSTELLUNG VON UMB: .....	14
INSTALLATION VON UMB: .....	14
AUTOEXEC.BAT .....	14
START.BAT .....	15
TNN176.PAS .....	16
SYSOP-Befehle: .....	21
(BE)ACON .....	21
(CL)EAR .....	22
(CONV)ERS .....	22
(DCD).....	23
(DOS) <KOMMANDO> .....	23
(DX)CLUSTER.....	24
(E)DIT .....	24
(K)ILL .....	24
(L3)MHEARD .....	24
(L)INKS.....	25
(LOA)D <FILENAME.EXT> .....	25
(M)AILBOX .....	25
(MH)EARD .....	26
(P)ARMS.....	26
(PAC)SAT.....	26
(PO)RT .....	27
(PR)OMPT .....	29
(RE)AD .....	29
(READB)IN.....	30
(RES)ET SYSTEM.....	30
(RES)ET <PORT> .....	30
(RU)BATCH.....	30
(S)TAT .....	30
(SP)ARM ODER BESSER SAVE P PARAMETER.....	30
(STAR)T <PROGRAMM>.....	30
(SUS)PEND .....	30
(SY)SOP .....	31

(TE)ST <PORT> .....	31
(TI)ME .....	31
(TR)ACE .....	32
TCPIP KOMMANDOS .....	33
(ARP) .....	33
(IPA)DRESS .....	33
(IPR)OUTE .....	33
ESC/ALT KOMMANDOS .....	34
DIE BESONDERE <ALT> X KOMBINATION .....	34
DIE VERSCHIEDENEN <ESC> TASTEN .....	34
(Hostmode-Befehle) .....	36
<b>Hostmode-Befehle .....</b>	<b>36</b>
<b>(#####.TNB) FILES .....</b>	<b>38</b>
(RUNBATCH) <FILENAME.TNB> .....	38
<b>EXTERNE PROGRAMME/BEFEHLE für den SYSOP .....</b>	<b>39</b>
(O)UTPUT .....	39
(CWER) .....	39
(CPERS) .....	39
(SETCALL) .....	40
(SYST)EXT .....	40
(SH)owSYS .....	40
(MSY) .....	40
(SYSH)ELP .....	41
(TOP) .....	41
(STARTCNT) .....	41
<b>Hilfsprogramme für den SYSOP zu Hause .....</b>	<b>43</b>
(MAKEDAT.EXE) .....	43
(TNNSET.EXE) .....	43
<b>Befehle für alle User .....</b>	<b>44</b>
(BE)ACON .....	44
(C)ONNECT .....	44
(CONV)ERS UND SEINE BEFEHLE .....	45
(CQ) .....	49
(D)EST .....	49
(DX)CLUSTER .....	50
(G)RAPH .....	50
(H)ELP .....	51
(L3)MHEARD .....	51
(L)INKS .....	52
(M)AILBOX .....	52
(MH)EARD .....	52
(N)ODES .....	53
(P)ARAMETER .....	56
(PAC)SAT .....	57
(PAC)SAT .....	57
(PI)NG .....	57
(PO)RT .....	57
(PO)RT * .....	58
(Q)UIT .....	59
(R)OUTES .....	59
(R)OUTES (V)ersion .....	60
(S)TAT .....	61
(TA)LK .....	64
(TI)ME .....	64
(U)SER .....	65
(V)ERSION .....	69
<b>EXTERNE Befehle für alle User .....</b>	<b>70</b>
<b>EXTERNE Programme für alle User .....</b>	<b>71</b>
DIGIMAIL: .....	71
(MSG) .....	71

LOCATORBERECHNUNG:.....	71
(QTH) .....	71
SATELLITEN-STANDORT-BERECHNUNG:.....	72
(SAT) .....	72
(TOP).....	72
VISITENKARTE:.....	73
(SAV)ECALL .....	73
(SH)OWCALL.....	74
ONLINEHILFE:.....	74
(H)ELP.....	74
<b>Framesammler .....</b>	<b>75</b>
<b>ANHANG A.....</b>	<b>76</b>
DIE VERSCHIEDENEN TEXTDATEIEN UND IHRE BEDEUTUNG.....	76
<b>ANHANG B.....</b>	<b>77</b>
TNC2C UND THENetNODE .....	77
KISS-SOFTWARE FÜR DEN TNC .....	78
WATCHDOG.....	79
<b>ANHANG C.....</b>	<b>80</b>
VANESSA .....	80
<b>ANHANG D.....</b>	<b>87</b>
(ETHERNET) ...WAS MUSS ICH TUN ?.....	87
<b>ANHANG E.....</b>	<b>90</b>
DER PACSAT-BROADCAST-SERVER VON TNN.....	90
<b>ANHANG F .....</b>	<b>92</b>
AUFBAU DER LAYER 3/4 FRAMES .....	92
<b>ANHANG G .....</b>	<b>97</b>
DIE GNU-VERSION VON TNN .....	97
<b>ANHANG H .....</b>	<b>100</b>
DAMA - EIN NEUES VERFAHREN FÜR PACKET-RADIO !.....	100
<b>ANHANG I.....</b>	<b>105</b>
TCP/IP UND TNN .....	105
<b>LIZENZ.....</b>	<b>110</b>
ALLGEMEINE LIZENZ FÜR AMATEURFUNK SOFTWARE (ALAS).....	110

### **Vorwort:**

TheNetNode wird von **Funkamateuren** für **Funkamateure** geschrieben, die sich bereit erklären, die nicht geringe finanzielle und zeitraubende Belastung auf sich zu nehmen, einen Packet-Knoten aufzubauen, zu betreiben und zu verlinken. TheNetNode erhebt keinen Anspruch darauf, FEHLERFREI zu sein. Aber die Softwareschreiber sind bemüht, die ihnen bekannten Fehler, die auch nachvollziehbar sein müssen, zu beseitigen. Deshalb gibt es über das Jahr verteilt immer wieder kleinere Updates. Weiteres bitte ich aus der ALAS am Schluß dieses Dokumentes zu entnehmen.

Die Programmversion für den Atari (ST) wird nun von Odo, DL1XAO, weitergeführt. Leider aber lassen sich die zukünftigen Erweiterungen von TNN dort nicht mehr alle realisieren. (z.B.: Einsatz von Vanessakarten o.ä.)

Die TheNetNode Version, für den TNC3 (von DK9SJ entwickelt) mit der 68302 CPU, wird nun von ODO DL1XAO gepflegt und beim Erscheinen einer neuen PC-Version kurzfristig zur Verfügung gestellt. Die Linux-Version von TheNetNode wird von Nils, DF6LN, gepflegt.

Die Besonderheiten der Linux-Version werden ausführlich in einer getrennten Dokumentation beschrieben, die zusammen mit dem TNN-Quelltext verteilt wird (doc/tnnlinux.doc). TheNet, einst für den TNC2 entwickelt, wird nicht mehr weiter gepflegt. Die heute benötigte Performance ist damit nicht mehr erreichbar.

TheNetNode ist somit auf den verschiedensten Hardwareplattformen zu Hause. Die damit verbundene Flexibilität dürfte jedem Betreiber eines Knotens entgegenkommen.

### **Das Programmpaket TheNetNode:**

TheNetNode ist ein Softwarepaket, in dem das gesamte Knotenprotokoll abgearbeitet wird, z.B.:

- Up - und Down-Link zu Usern mit oder ohne DAMA-Protokoll,
- Interlink-Protokoll mit Routing,
- Benutzeroberfläche (Hilfen und Informationen),
- Einbinden verschiedener Schnittstellen,
- Zugriff auf externe Programme wie QTH- und SAT-Berechnungen, PR-Callbook,

um nur einiges zu nennen.

Dem Programm stehen für die Kommunikation mit der Außenwelt verschiedene Schnittstellen zur Verfügung:

- Serielle I/O mit Tokenring Protokoll (9k6 bis bestenfalls 115kBit),
- Maximal 4 Serielle I/O (bei Linux maximal 16) mit Kisslink-Protokoll (ähnlich Tokenring),
- Parallele Schnittstelle zum Frontendrechner Vanessa (bis 140 Kbit/s je Interlink),
- Parallele Schnittstelle zum Ethernet (10 MBit/s) mit externen Treibern möglich,
- BayCom USCC mit dem internen Treiber.

### **BAYCOM SCC-Karten:**

TheNetNode unterstützt über einen integrierter SCC-Treiber die Karten USCC, OptoSCC und DSCC (Digi-SCC BayCom). Es ist nur möglich **EINE** Karte mit maximal 8 Ports anzusteuern.

Hierzu muß im Enviroment eingetragen werden:

SET USCC=base\_hex,irq\_dez für USCC(BayCom) oder  
SET DSCC=base\_hex,irq\_dez für DSCC(BayCom) oder  
SET OSCC=base\_hex,irq\_dez für OptoSCC

In TheNetNode kann dann die jeweilige Hardware (USCC0...x, OSCC0...x, DSCC0.xx) an einen beliebigen Port gebunden werden:  
PORT 4 USCC2

Die Konfiguration erfolgt über den Mode-Parameter.

DF9IC -> MODE=9600rtz

AFSK -> MODE=1200c

Der SCC-Treiber ist kein neues Feature, läuft aber endlich vernünftig.

### **Der Tokenring**

Um über eine Rechnerschnittstelle mehrere HF-Ports bedienen zu können, wurde der Tokenring eingeführt, der auch über etliche Jahre eine preiswerte und ausreichende Schnittstelle zur Verfügung stellte.

Die TNC bedienen mit einer speziellen KISS-Soft die Funkseite, als da sind Sendertastung, Daten senden, Daten empfangen, zum Rechner weiterleiten und von ihm empfangen.

Für die Weiterleitung wird ein Tokenring eingesetzt. Der Rechner sendet ein sogenanntes Token (bestimmte Bytefolge) auf den Ring. Da alle TNC in einer Reihenschaltung hintereinander angeordnet sind, kommt es bei dem ersten TNC im Ring an. Hat der nun auf der Funkseite etwas empfangen, so hält er das Token zurück, sendet statt dessen die soeben empfangenen Daten an seinen Ringnachbarn und hängt an diese Daten das Token wieder an. Da jeder TNC immer auf das Token wartet, bevor er Daten auf den Ring legt, läßt jeder Nachbar-TNC die Daten vom vorhergehenden TNC unverändert passieren. Sowie er aber das Token erkennt, hält auch er es zurück und legt seine eigenen Daten auf den Ring und dann erst wieder das Token. Irgendwann, je nach Anzahl an TNC, kommen die Daten nun beim Rechner an.

Hier wird nun das TheNet-Protokoll abgearbeitet und die Daten werden, mit einer Port-Adresse versehen, wieder an ein Token angehängt und vom Rechner auf den Ring gesendet. Die TNC prüfen aber auch, ob bei den Daten, die sie empfangen, welche dabei sind, die an sie bestimmt sind (Portnummer). Diese werden von dem Token abgehängt und auf der Funkseite mit den entsprechenden Parametern ausgesendet. Das ganze Verfahren scheint auf den ersten Blick einiges langsamer zu sein, oder auch nicht schneller als das Vernetzen der TNC untereinander mit Diodenmatrix. Dem ist aber nicht so! Man kann eine deutlich spürbare Geschwindigkeitssteigerung feststellen! Woran liegt das? Einmal daran, daß der Tokenring immer zu annähernd 100% ausgelastet ist, Kollisionen zwischen den TNC nicht möglich sind und keiner der TNC Warteprozeduren ausführen muß, um Kollisionen zu vermeiden. Aber der Hauptgrund der Geschwindigkeitssteigerung liegt darin, daß der Z80 im TNC mit Hilfe des KISS-Programmes bei weitem weniger zu tun hat als mit dem kompletten TheNet. Dieses Programm besteht fast nur aus Interruptroutinen (z.B. Umsetzung Synchron- auf Asynchron-Übertragung), welche die SIO bedienen und die Pakete zwischen V24 und Funkschnittstelle verteilen. Es wird keine unnötige Rechenzeit damit vergeudet, sich um den Inhalt der Pakete zu kümmern. Dazu ist ja der Atari-ST bzw. der PC da, der das bei weitem schneller kann. Die Aufarbeitung der Pakete und die Behandlung der einzelnen AX25-Protokollebenen läuft auch deshalb weit schneller, weil sie trotz mehrerer angeschlossener TNC nur einmal erfolgt und nicht, wie vom TNC-TheNet gewohnt, in jedem TNC separat für sich.

Die TNC sollten mit 9,8304 MHz Quarzen und 10 MHz SIO + CPU ausgestattet sein.

Im Übrigen wird die KISS-Software (allerdings ohne die Token-Ring-Erweiterung) schon seit langem bei den WAMPES-Knotenrechnern, dem TCP/IP Packet von KA9Q und auch dem Packet-Terminalprogramm SUPERKISS mit Erfolg verwendet.

Auf der Schnittstelle, beim PC muß sie nun mit SET TOKENCOM=<com Nr.> definiert werden, sind Übertragungsraten von 9.600, 19.200, 38.400 Bit/Sec möglich. Mit dem FIFO-Schnittstellenbaustein 16550 AFN von National als RS232 auch darüber hinaus noch 57.600 und 115.200 Bit/Sec. Das Programm erkennt diesen Baustein selbständig (Anzeige beim Hochfahren des Rechners und auch bei Abrufen der Statistik ersichtlich) und wendet diesen Mode an. Vereinfachte Arbeitsweise: Es werden in dem 16550 AFN bei TNN bis zu 8 Byte zwischengespeichert und erst dann gibt es für den Rechner einen Interrupt. Intern im Chip ist natürlich ein Timer, der den Datenfluß überwacht und auch ggf. mal z.B. 3 Byte weiterreicht.

Vorteil des Ringes ist, jeder TNC sendet nur an EINEN anderen Empfänger oder andersherum betrachtet, jeder TNC empfängt nur von EINEM anderen Sender Daten. Das bedeutet, das Pegelprobleme, wie sie beim Parallelbetrieb von mehreren TNC auf einer gemeinsamen Diodenmatrix auftreten, hier nicht vorkommen.

Doch Vorteile haben meist auch Nachteile. Fällt in einem Ring auch nur ein TNC aus, so fällt, wenn man keine Vorsorge trifft, der Ring und somit der gesamte Knoten aus. Ein weiterer Nachteil ist, daß sich die Baudrate des Tokenring auf die einzelnen Funkstrecken aufteilt wird.

### **Bei Verwendung der GNU32 Programme sind FIFO-Bausteine in den Comschnittstellen PFLICHT!**

Informationen zur TNC-Software und zu Watchdog siehe Anhang B.

### **Der Kisslink**

Da immer wieder Drahtanbindungen zu anderen Systemen wie Mailboxen, Cluster oder anderen Knoten geschaffen werden müssen, verfügt TheNetNode auch über eine KISS-Schnittstelle. Dieses KISS-Protokoll darf aber nicht mit dem anfangs beschriebenen speziellen TOKENRING-KISS verwechselt werden. Die KISS-Schnittstelle kann mit SET KISS1=1 bis SET KISS4=4 einer COM-Schnittstelle zugewiesen werden. Siehe AUTOEXEC.BAT und (PO)rt-Befehl. Als KISS-Treiber für andere Systeme sind z.B. TFKISS, TFPCR und TFPCX zu nennen. Auch der Anschluß eines TNC2c mit einem KISS-/SMACK-EPROM ist möglich.. Über die (MO)de Funktion beim (PO)rt-Befehl lassen sich unterschiedliche CRC-Verfahren einstellen. Aber auch hier nochmals der Hinweis auf den Schnittstellenbaustein 16550 AFN.

### **Bei Verwendung der GNU32 Programme sind FIFO-Bausteine in den COMschnittstellen PFLICHT!**

### **Die Vanessakarte**

Bei der Vanessakarte handelt es sich um Slot-Karten, die mit einer schnellen CPU und SCC ausgestattet sind. Als Schnittstelle zum Rechner benötigt sie einen ISA-Slot und zur Funkseite können z.B. RS 422 Schnittstellentreiber eingesetzt werden (Stromschnittstelle), die den abgesetzten Betrieb eines Modems über mehrere 100 Meter ermöglichen.



Durch die parallele Datenverarbeitung auf den Bus des Rechners hat es einen mächtigen Ruck im Datendurchsatz gegeben. Die Verarbeitungsgeschwindigkeit des Knotens wird nun durch seine Links (Baudrate und das Übertragungsverfahren (Semi - oder Vollduplex)) bestimmt. Die Laufzeiten innerhalb des Knotens sind durch die parallele Verarbeitung der Daten so gering geworden, daß sie auch bei einem 64 Kbit/s Vollduplexlink nicht ins Gewicht fallen. Bei externem Takt (also synchroner Datenverbindung) kann die Vanessakarte bis zu 140 Kbit/s je Port.

Auf einer Vanessakarte befinden sich 2 Ports.

Ausführliches zur Vanessakarte siehe **Anhang C**.

Bei Henning, DG9FU, kann ein Teilesatz bestehend aus: Leiterplatte, CPU und SCC bezogen werden.

Für den Anschluß eines G3RUH-Modem ist eine kleine Adapterplatte nötig, die wieder die Umsetzung von RS422-Pegel auf TTL-Pegel vornimmt.

Unterlagen (Beschreibungen und Schaltungen) können bei Henning, DG9FU, gegen Erstattung der Unkosten in Höhe von DM 10.- (Fotokopien 6.- und Porto 4.-) angefordert werden. Adresse auf der letzten Seite.

### **Externe Treiber**

In der DOS- und GNU-Version sind die Schnittstellentreiber für VANESSA, KISSLINK, Baycom USCC und TOKENRING in der Software integriert. Anders sieht es bei Ethernet aus. Für diese Anwendungen wurde die TNL1.EXE geschrieben das den Kontakt zwischen TNN und den EXTERNEN-Treibern herstellt. Als Treiber können nun die Kartentreiber die auch PC-Flexnet verwendet zum Einsatz kommen.

Ab der Version V1.73 ist es möglich, externe Treiber vor TNN zu laden. Diese werden dann automatisch erkannt und eingesetzt.

Zur Nutzung der externen Treibern in TheNetNode muß zuerst ein Programm geladen werden, welches die Kommunikation zwischen TNN und Treiber ermöglicht. Dieses Programm (TNL1.EXE) muß immer vor den Treibern geladen werden. Ausnahme: Treiber für z.B. Netzwerkkarten müssen vorher geladen werden. Diese setzen auch nicht auf den TNL1 auf und sind meist herstellerspezifisch.

TNL1 enthält ein 16Bit- und ein 32Bit-Interface, d.h. man kann mit TNL1 auch Treiber laden, die von der GNU32-TNN angesprochen werden können.

TNL1.EXE kann für mehr Informationen mit -v aufgerufen werden und ein paar Parameter angegeben werden.

TheNetNode Layer1 Interface V5.2 (c) 1996 by NORD><LINK (DB7KG,DG1KWA)  
Resident portion loaded, handler installed to IRQ 0x60  
32-Bit Protected Mode handler installed to IRQ 0x61  
Manifest is at 222A:0000  
511312 bytes free memory  
2432 bytes CODE  
2672 bytes static DATA  
528 bytes STACK

Usage: tnl1 [-?] [Options]]

-h -?        diese Hilfe anzeigen / give this help  
-32         32bit Interface abschalten / disable 32bit interface  
-u         TNL1 und Treiber entladen / unload tnl1  
-v         ausführliche Meldungen / verbose

Der Ablauf für das Starten eine TNN mit externen Treibern könnte z.B. so aussehen:

```
; hier einen evtl. Packet Driver für die Ethernet-Karte laden  
;  
NE2000 0x65 10 0x300  
;TNL1 laden  
TNL1.EXE  
; AXIP-Treiber laden  
; ein NE2000-Treiber muß geladen sein !  
IPPD -i:0x65 -m:44.130.88.1 -p:44.130.88.2  
; TheNetNode  
TNN173.EXE  
; alles entladen  
TNL1.EXE -u
```

Die Treiber für VANESSA, KISSLINK (bis zu 4 Stück) und TOKENRING sind weiterhin fest in TNN eingebunden und sollten den externen Treibern (nur VANESSA und KISSLINK) vorgezogen werden, da die internen Treiber wesentlich schneller sind und einige Verbesserungen bieten (z.B. KISS-SMACK oder externer Clock bei der VANESSA). Nach dem Start erkennt TheNetNode automatisch die Treiber, sie müssen dann nur noch auf die entspr. Baudrate und Mode konfiguriert werden.

### **Port-Nummerierung**

Da die Treiber von Port 0 an aufsteigend angesiedelt werden, kann es Probleme mit anderer Hardware z.B. VANESSA geben. Deshalb kann als Platzhalter ein Dummy-Treiber geladen werden. Mit diesem ist es möglich, eine anzugebende Anzahl von Ports freizuhalten.

Syntax : DUMMY <Port-Anzahl>

Diese zwei Anweisungen

DUMMY 6

IPPD -i:0x65 -m:44.130.88.1 -p:44.130.88.2

bedeuten, daß 6 Ports freigehalten werden und dann der AXIP-Treiber geladen wird.

### **Welche Treiber werden verwendet**

Zur Zeit werden die Treiber des PC/FlexNet verwendet (aktuelle Version). ACHTUNG: Die Treiber der PC/FlexNet 3.3c gehen nicht mit TNN !

### **GNU32-TNN**

Die externen Treiber funktionieren inzwischen auch sehr gut mit der GNU32Bit-Version von TheNetNode, allerdings gibt es hier eine Sache zu beachten. Geräte die viele IRQ erzeugen, sind für eine GNU-Version Gift, da sie die CPU zwingen, zwischen Real-Mode und Protected-Mode laufend hin und her zuschalten. Dieses Umschalten kostet aber viel an Rechnerpower.

Treiber wie SER12 oder PAR96 laufen sicher nicht ohne Probleme. Bei einer reinen DOS-Version von TheNetNode sollte es keine Probleme geben.

Bei den Netzwerktreiber sieht die Sache etwas besser aus, diese sollten ohne Probleme laufen. In einem Test lief sogar die GNU32-Version schneller (!) als eine DOS-Version !!!

Folgende Treiber gibt es zur Zeit, die Liste erhebt aber keinen Anspruch auf Vollständigkeit !

IPPD	AXIP-Treiber; damit kann man seinen LINUX-Rechner (WAMPES oder DP-BOX) an TNN anbinden. Läuft mit GNU- und DOS-TNN
IPXPD	Netzwerktreiber mit IPX-Protokoll, setzt auf einen geladenen PacketDriver auf. GETESTET ....geht UFB !
IPXN	IPX in einem Novell-Netzwerk NICHT GETESTET !
KISS	KISS-Treiber, der interne TNN-Treiber sollte aber vorgezogen werden Läuft mit DOS und GNU
VANESSA	Vanessatreiber; der interne TNN-Treiber sollte aber vorgezogen werden, da er wesentlich schneller ist! Läuft mit DOS und GNU

Für alle Treiber gilt: vor Einsatz die mitgelieferte DOK lesen !!! Wenn jemand TheNetNode mit einen der Treiber verwendet, würden wir uns über etwas Feedback sehr freuen !

### **Ethernet-Schnittstelle:**

TheNetNode(PC) bietet die Möglichkeit, mehrere TNN oder Mailboxen per 10 MBit/s Ethernet miteinander zu koppeln. Erforderlich sind lediglich preiswerte Ethernet-Karten und ein passender Treiber nach der TCP/IP PKTDRV-Spezifikation.

Nun braucht man nur ein dünnes RG58 Kabel zu verlegen und 2 Stück 50 Ohm Abschlußwiderstände oder ein Twisted-Pair und kann dort beliebig viele Rechner anschließen. Neben weiteren TNN-Rechnern z.B. auch eine Mailbox oder einen DX-Cluster (Treiber in Arbeit) usw..

Für Mailbox oder Useranbindung:

Das Programm „TFX\_NET.COM“ verhält sich wie ein TFKISS-Treiber, was die Schnittstelle zum Terminalprogramm angeht. TFX\_NET wird einfach als residentes Programm installiert, nach dem der PKTDRV-Treiber geladen wurde. Somit kann man z.B. Terminalprogramme wie TOP, GP, THP und sogar WinGT betreiben und den Knotenrechner über das Ethernet-Netzwerk connecten!! So etwas ist z.B. prima für abgesetzte Sysop-Terminals im QRL.. Natürlich geht dies genauso auch mit der DIEBOX-Mailbox und anderen Programmen, welche die TFPCR-Softwareschnittstelle unterstützen. Da die AX25 Daten einfach in Ethernet-Frames „verpackt“ und im Kabel „gebroadcastet“ werden, können sich alle Teilnehmer auch untereinander connecten.

Die Vorteile liegen auf der Hand:

- Ethernet-Karten (meist NE2000 kompatibel) sind sehr preiswert und kosten höchstens 1/3 oder weniger als einfache TNC.
- Doppel-TNCs oder VANESSA/TNC-Kombinationen können entfallen, dadurch spart man einiges an Finanzen und Aufwand ein.
- Der KISSLINK mit all den Problemen der PC-RS232-Schnittstellen kann entfallen.
- es reicht ein „dünnes“ RG58 oder Twisted-Pair quer durch alle Räumlichkeiten, wo gerade Platz ist.
- galvanische Entkopplung.

Mehr dazu im ANHANG D.

### **DPMI oder Nutzung des meist brachliegenden Speichers über 1MB:**

Unter MS-DOS hatten wir bisher immer damit zu kämpfen, daß der verfügbare Speicher auf 1 MB (+ HMA) beschränkt war. Diesen Speicher mußten sich alle Programme teilen. Damit waren Programme wie TNN in einen Speicherbereich eingezwängt. Mit dem Dos-Protected-Mode-Interface stehen bis zu 16 MB bzw. 4 GB theoretischen Speicher zur Verfügung. Aber warum noch MS-DOS unterstützen, wo es so gute Betriebssystem wie OS/2, LINUX oder WIN 95 gibt ? Alle diese Betriebssysteme haben einen recht hohen Speicherbedarf, außerdem schleppt man bei WIN 95 eine für unsere Zwecke unnötige Oberfläche mit. Außerdem wird sich nicht sofort jeder Sysop mit LINUX anfreunden können - aber es lohnt sich trotzdem, sich damit zu befassen.

Deshalb die weitere Unterstützung von MS-DOS.

Da die GNU-Version eine Erweiterung der „normalen“ DOS-Version ist, habe ich die Erklärungen hierzu in den **ANHANG G** verlagert.

### **TheNetNode-Programm:**

Damit sind wir eigentlich beim Programm angekommen. Zu erwähnen sei noch, daß auch der Rechner von einem Watchdog auf der Schnittstelle überwacht und notfalls neu gestartet wird, denn auch das beste Programm kann sich mal irgendwohin verirren....

Auf der Diskette befindet sich das Programm TNN176.EXE. Paßwort, IDENT (Rufzeichen), ALIAS usw. sind im File TNN176.PAS abgelegt, das nach Möglichkeit nicht mehr verändert werden soll. Die mit (S)tat abrufbaren statistischen Werte werden je nach Einstellung des Parameters SaveConfig in X mal 10 Minutenabständen, also bei der Einstellung '6' jede Stunde einmal, auf der Festplatte in dem File TNN176.STA abgespeichert. Somit kann maximal noch eine Stunde in der Statistik je Neustart fehlen. Wie lange das Programm ohne Neustart in Betrieb ist, kann aus der ersten Zeile der Statistik ersehen werden. Die Parameter werden nun über die TNN176.TNB eingestellt. In ihr sind alle benötigten voreingestellten Parameter enthalten. Beim Programmstart sucht TNN.EXE nach den Files, TNN176.PAS, TNN176.STA und TNN176.TNB und lädt daraus seine Betriebsparameter in den Rechner. Sind die Files nicht vorhanden oder defekt, so wird beim Start am Monitor angezeigt :

WARNING: Fehler in TNN176.PAS \*\*\*  
\*\*\* TNN176.STA - Open-Error \*\*\*  
WARNING: Fehler in TNN176.STA \*\*\*

ACHTUNG: Packet-Treiber (PKTDRV) fehlt !!

Login: TNN176.TNB not found !

Das Programm benutzt dann die Default-Parameter und legt die Files TNN176.PAS und TNN176.STA neu an. Sind die Files in Ordnung, erfolgt kein besonderer Hinweis.

ACHTUNG: Die Meldung „Packet-Treiber fehlt“ ist NUR der Hinweis, daß eine Ethernet-Schnittstelle NICHT angesprochen werden kann. Sie hat aber für den Betrieb OHNE Ethernet keine weiteren Nachteile !

Nach dem Ändern von Parametern wird TNN176.STA jeweils nach Verlassen des Sysopmodus durch Quit, Disconnect oder Connect jeweils neu auf die Festplatte geschrieben. Damit ist sichergestellt, daß bei einem Programmneustart die Parameter wieder richtig gesetzt werden.

Die Files TNN176.PAS und TNN176.STA haben die Versionsnummer im Filenamen. Damit ist es dann auch möglich, unterschiedliche Programmversionen im Knoten abzulegen.

TheNetNode ist im Moment auf 400 Level 2 und 200 Level 4 Verbindungen eingestellt. Dieses sind aber keine Endwerte. Die Verwaltung von einer größeren Menge an Verbindungen ist nur noch eine Frage des Speichervolumens des Knotenrechners und seiner Taktfrequenz.

### **DAMA ... Verkehrsregelung durch den Knoten:**

DAMA ist heute auf den Userzugängen eine weitverbreitete Betriebsart. Es ist nun nur noch der entsprechende Port <nr> DAMA=on zu setzen. Parametereinstellungen sind keine mehr nötig.

Im Port-TNC auf dem DAMA laufen soll muß das DAMA-Bit im Eprom gesetzt sein. Bei gesetztem DAMA-Bit sendet der TNC an den Knotenrechner die Information zurück, wenn ein Paket an einen User abgesendet wurde. Diese Information wird für die Steuerung der DAMA-Funktion benötigt. **(Bei der Vanessakarte nicht erforderlich.)**

### **DAMA-Ablaufsteuerung:**

Stationen, die keine DAMA-fähige Usersoftware benutzen und OHNE Aufforderung durch den DAMA-Master senden, werden IM QSO verwarnt. Sind mehr Verwarnungen nötig, als unter DAMA-MaxPol angegeben, so wird der User disconnected. Es dürfen auch **NICHT mehrere** TNC unter gleichem Call aber unterschiedlicher SSID betrieben werden, da der Knoten im DAMA-Modus alle bestehenden Links zum Senden auffordert. Zwei TNC mit gleichem Call aber unterschiedlicher SSID würden gleichzeitig senden !

Für den TNC2 gibt es ein DAMA-fähiges Eprom, es steht in der Mailbox unter der Rubrik 'TNC2'. Dort wird auch meist ein kostenloser Eprom-Service angeboten. Wer die Vorteile von DAMA nutzen möchte, sollte unbedingt **The Firmware 2.7b** nachrüsten! Seit der **The Firmware 2.7b** kommt es auch bei Multiconnect nicht mehr zu Meckermeldungen. Der Softwarefehler, daß der TNC nach Anpollen einer DAMA-Verbindung die anstehenden Daten aller bestehenden Verbindungen gesendet hat, ist beseitigt.

Alle anderen Benutzer sollten unbedingt FRACK auf den Maximalwert einstellen.

**\*\*\* Wichtig \*\*\***

Auf dem DAMA-Einstieg sind direkte QSO unter Umgehung des Knotens zu unterlassen, da sie, wie jede nicht vom Master angeforderte Aussendung, zur Störung des gesamten Betriebes führen !!! Dies gilt auch für Betrieb über eine andere Station als „Digipeater“, um den Knoten auf der gleichen Frequenz zu erreichen.

L2 Digipeating auf einem DAMA - Port führt nun zu einem sauberen Eintragen der Linkverbindung und damit zu einem reibungslosen DAMA - Verkehr.

USER mit Multiconnect kommen gegenüber USERN mit nur einer Verbindung zum Knoten nicht öfters an die Reihe. Aber „Leerpolls“ werden nun auch entsprechend beachtet.

### **SYSOP-Arbeiten am Knoten:**

Bedienen kann man das Programm einmal von der Konsole (Tastatur) aus oder auch über Funk. Für beide Fälle ist eine Identifikation als SYSOP notwendig. Von der Konsole aus wird der Benutzer aufgefordert das „Paßwort“ einzugeben. Als Default-Paßwort ist „Geheim“ im Programm festgelegt. Ist es korrekt eingegeben (Groß- und Kleinschreibung beachten), wird man mit

Welcome to TheNetNode (PC), Version 1.76 (Datum)

begrüßt. Mit <ESC> c <RETURN> kann man sich nun als HOST mit dem Knoten verbinden. Als HOST ist man auch automatisch SYSOP und kann alle Änderungen an den Parametern vornehmen.

Nach Ende der Arbeiten kann man ihn wieder mit <ESC> logout <RETURN> verschließen. Ein automatisches Ausloggen findet vor jedem Stundenübergang statt.

Von der Funkseite ist der Knoten erst zu connecten und dann der Sysop-String mit mindestens „sy“ abzufragen. Daraufhin bekommt man folgenden Text zugesandt: „KS:DB0EAM> 62 36 65 13 34“. Diese Ziffern geben jeweils die Position eines Zeichens im Paßwort an. Mit den zu den Ziffern gehörenden Zeichen muß nun geantwortet werden. In dem Paßwort dürfen KEINE Leerzeichen vorkommen und die angeforderten Zeichen müssen direkt nacheinander folgen. Vor und nach den angeforderten Zeichen dürfen auch andere Zeichen stehen, so daß nicht mehr ohne weiteres ersichtlich ist, welche Zeichen die Antwort darstellen. Eine Quittung, ob die zurückgesendete Zeichenkette richtig oder fehlerhaft war, kommt nicht. Diesen Vorgang kann (und sollte) man mehrmals hintereinander durchführen, wobei er nur einmal richtig sein muß. Ob dann der 1. 2. oder 3. Versuch das richtige Paßwort enthielt, spielt keine Rolle. Die mehrfache Identifikation dient nur der Verwirrung der „Geister“, die allzugern auch das Paßwort hätten, um dem Sysop „hilfreich“ unter die Arme zu greifen. Ein wahrer Sport soll sich da mancherorts schon entwickelt haben, hihi. Die meisten Terminalprogramme für Packet Radio unterstützen die automatische Privilegierung als Sysop.

Um den hilfreichen „Geistern“ auch den letzten Zahn zu ziehen, so steht zumindest bei mir, wenn ich das Programm zum Knoten als Binärfile fernlade, stets das Paßwort auf den Default-Parametern „12345678901234 usw.“. Damit ist nun auch das Mitschreiben des Binärfiles sinnlos. Einerseits kann das Programm jeder bekommen, andererseits kommt man auch auf diesem Weg nicht an das Paßwort heran. Das aktuelle Paßwort steht ja bereits in TNN176.PAS.

In der Datei SYSOP.PRO wird zusätzlich noch ein Logbuch geführt, das jede Eingabe des SYSOP-Befehls registriert. Es werden Rufzeichen, Datum und Uhrzeit gespeichert. Diese Datei kann vom SYSOP ausgelesen und bei Bedarf auch wieder gelöscht werden, wenn sie zu lang wird.

Nun sollte zumindest dem Sysop die „gewaltige“ Herrschaft über den Knoten offen stehen.

### **Noch ein sehr wichtiger Punkt zu TheNetNode :**

Auf dem Knoten wird eine MH-Liste geführt. Sie wird als MHEARD.TAB auf dem aktuellem Laufwerk gespeichert und bei JEDEM User-Paket im RAM aktualisiert. Soll als aktuelles LW eine Floppydisc benutzt werden, so sind Probleme vorprogrammiert.

**Ausweg:** Es wird auch in der Zukunft immer mehr Speicherraum für die Zwischenspeicherung von Daten benötigt, z.B.: Führen einer Namenliste und evtl. eine QRV-Liste. Das Auslesen der `U +` Liste wird als .TMP-File auch zwischengespeichert. Auch dieses soll nach Möglichkeit ohne Zeitverzögerung geschehen. Es bietet sich also nahezu an, eine kleine Harddisk mit in den Knotenrechner einzusetzen. Wer das im Moment nicht kann, muß sich mit einem virtuellen Laufwerk (RAM-DISC) behelfen. Hierhin sollten dann alle .TXT-Files kopiert werden und auch evtl. COMMAND.COM. Sie muß noch den nötigen Platz bieten, um das MHEARD.TAB File zu führen (Größe richtet sich nach der Einstellung MH = ...) und auch die Liste `U +` noch aufnehmen zu können. Ist der Platz nicht vorhanden, so ist ein Programmabsturz vorprogrammiert !

### **Rechnerkonfiguration: (Hier als Beispiel für einen 386 oder 486 mit HD)**

```
CONFIG.SYS
*****
DEVICE=c:\HIMEM.SYS
DEVICE=c:\emm386.exe noems x=d000-dfff i=e000-f3ff
DOS=HIGH,umb
COUNTRY=049,850,C:\DOS\COUNTRY.SYS
FILES=30
BUFFERS=40
STACKS=9,256
```

**EMM386 bzw. QEMM zur Bereitstellung von UMB:****Upper Memory Support [UMB] 2.0 für TheNetNode 1.70 by DB7KG**

Erst einmal, was sind UMB?

UMB nennt man den Frei-Speicher oberhalb 0A0000 bis maximal 1 Mb, er wird von verschiedenen 386 Speichermanagern zu Verfügung gestellt. Er dient dazu, Treiber aufzunehmen, damit diese nicht den Speicher unterhalb 640 KByte belasten.

Hier die Features der neuen Version:

- 1.) autokonfigurierend:  
Einfach nur QEMM oder EMM386 installieren und TNN starten.
- 2.) Benutzung ALLER freien UMB-Blöcke,  
nicht nur EINES Blockes. Das kommt besonders EMM386-Usern zugute, da EMM386 den UMB-Speicher ziemlich zerstückelt

**Installation von UMB:**

Als erstes ist ein UMB-Treiber wie EMM386 oder besser QEMM zu installieren. Bei EMM386 ist eine umfangreiche Konfiguration nötig, QEMM ist weitgehend selbstkonfigurierend.

Tip für EMM386:

- 1.) Je nach verwendeter Grafikkarte ( Monochrom oder VGA ) können unterschiedliche Bereiche mit I=xxxx-xxxx beim Aufruf des EMM386 freigegeben werden. A000-AFFF; B800-BFFF; F000-F7FF bzw. F000-FAFF. Der Bereich C800-CFFF kann, wenn keine HD benutzt wird, auch freigegeben werden. Hier muß jeder auf seinem Rechner selbst testen!
- 2.) E000-EFFF meistens benutzbar, ausprobieren. C000-C7FF fast immer nutzbar.
- 3.) D000-DFFF MUSS jedoch bei dem Einsatz von Vanessakarten gesperrt werden. (Bereich je nach Adressierung der Vanessakarten !)

Als Hilfe hier noch ein paar Bereiche, die evtl. nutzbar sind:  
A000-AFFF : Grafikspeicher im Grafikmodus,  
B000-B7FF : Grafikspeicher (monochrom),  
B800-BFFF : Grafikspeicher (NICHT monochrom),  
C000-C7FF : ROM der VGA-Karte,  
F000-F3FF : ROM-Bios.  
(Quelle: PC-Welt 3/94)

Eine Übersicht über den verwendeten Speicher bekommt man mit dem Diagnoseprogramm MSD.EXE. Es zeigt die belegten und verfügbaren Speicherbereiche an.

Auch das Programm MEMMAKER.EXE ab MSDOS 6.2 gibt selbständig die verwendbaren Speicherbereiche frei.

**II. Support von TNN**

Einfach TNN starten, alle UMB-Blöcke werden automatisch erkannt und benutzt.

ACHTUNG : UMB gibt es bei GNU, LINUX und TNC3 natürlich NICHT !!!!!

Das Auskommentieren eines Bereiches mit x=D000-DFFF beim EMM386 ist nur bei dem Einsatz der Vanessakarten notwendig. Bereich je nach Adressierung der Vanessakarten.

Die Speicherverwaltung kann vom Sysop über das DOS-Programm MEM abgefragt werden. (Siehe DOS-Dokumentation).

**AUTOEXEC.BAT**

Übersichtshalber habe ich nun die AUTOEXEC.BAT aufgeteilt und für die TNN-Spezifischen Einstellungen eine START.BAT angelegt. Die AUTOEXEC.BAT enthält die rechnerspezifischen Einstellungen, die START.BAT die Einstellungen für TheNetNode.

```
@ECHO OFF
prompt=$p$g
lh KEYB GR,,a:\KEYBOARD.SYS
path c:\dos; c:\tnn
START.BAT
```

### START.BAT

```
set tnncfg=1009,32
rem set tnnbuffers=1500 (für einen 286)
rem set tnnbuffers=2000,UMB (für 386 und 486)
rem set tnn32buffers=5000 (für die GNU-Version)
set conversd=P:0
set TZ=UTC0
set msgpath=C:\TNN\MSG\
set QTH=<koordinaten> oder <locator>
set CONSOLE=3
set TOKENCOM=2
set KISS1=1
set copycmd=/y
b-log.exe
cd TNN
TNN176.exe
```

### Erklärungen zur START.BAT:

#### SET TNNCFG=<1.Anzahl>,<2.Anzahl>

Die 1.Anzahl gibt die Größe der Nodes / Destinations Liste an. Default Einstellung :1009.

Die 2. Anzahl ist die Länge der Linkliste. Default : 32.

Die Verwaltung der Routing-Tabellen von TheNetNode erfolgt mit einem sogenannten Hash-Algorithmus. Dieser arbeitet sehr effizient, solange die Nodesliste zu weniger als 50% gefüllt ist. Ist der Füllungsgrad höher, erhöht sich die Zugriffszeit auf die Tabelle überproportional hoch. Auf Systemen mit wenig Speicher sollte die Nodes-Tabelle auf mindestens 30% mehr als die zu erwartende Nodes-Anzahl eingestellt werden. Auf Systemen mit viel Speicher kann die Liste ruhig um Faktor 2-3 überdimensioniert werden.

#### SET TNNBUFFERS=

**NUR FÜR DIE DOS-VERSION !** Über die Umgebungsvariable TNNBUFFERS kann TNN mitgeteilt werden, wieviele Buffer vom konventionellen Speicher (640 KByte) benutzt werden sollen. Sie wird in der START.BAT mit z.B. „SET TNNBUFFERS=2000“ gesetzt. Beim Betrieb mit einem „386 oder 486“ kann zusätzlich zum den Buffern im konventionellen Speicher über den Zusatz **UMB** der Zugriff auf den oberen Speicher zwischen 640 KByte bis 1 MByte freigegeben werden.

TNN läßt in jedem Fall 64KB Arbeitsspeicher für externe Programme frei.

Die Auswirkungen nach der Änderung dieser Einstellung auf die Buffer-Verwaltung lassen sich nach dem Starten von TNN (im connecteten Zustand), mit dem Befehl DOS-MEM /c ablesen.

#### SET TNN32BUFFERS=

**NUR FÜR DIE GNU-VERSION !** Ansonsten gleiche Wirkung.

#### SET CONVERSD=P:0

P bestimmt, ob das Protokoll auf Kanal 32767 angezeigt wird. Das Protokoll zeigt eingehende und ausgehende Hostbefehle an und kann zu Analyse Zwecken eingesetzt werden.

P:0 (default): Protokoll aus

P:1 Protokoll an

#### SET TZ:

Die Variable TZ wird auch für den Conversmode und die SAT-Berechnung benötigt. Damit die Zeiten stimmen, ist sie, je nach eingestellter Rechnerzeit, auf UTC0, MEZ-1, EST-2 oder CET-1 zu stellen.

#### SET MSGPATH:

Pfad für MSG-Files

#### SET QTH=<koordinaten> oder <locator>:

Mit der Variablen QTH werden für das Programm QTH.EXE und SAT.EXE im Verzeichnis USEREXE die Koordinaten oder Locator des Knotens eingegeben. Im Nahbereich sind die Koordinaten natürlich weitaus besser geeignet. Siehe: externe Programme für alle User.

**SET CONSOLE=<com Nr.>,<Adresse>,<irq>**

Die Schnittstelle für die CONSOLE kann auf COM 1 oder auf COM 2 gelegt werden. Die Angaben <Adresse> und <irq> sind nur für exotische Adressen und Interrupts notwendig. Wird die Console durch HOSTMODE belegt, so steht die Tastatur jedoch nicht mehr zur Verfügung. Mit <ESC> I <call> muß dann das Call des angeschlossenen Dienstes eingetragen werden. Das Call kann dann auch mit M <call> oder DX <call> verbunden werden damit es in den Nodes - / Destinations – Listen eingetragen wird. Das Host - Interface wird über Interrupts gesteuert. Der Speed ist derzeit auf 19200 Bit/s festgelegt.

**SET TOKENCOM=<com Nr.>,<Adresse>,<irq>**

Die Schnittstelle für der TOKENRING kann auf COM 1 oder auf COM 2 gelegt werden. Die Angaben <Adresse> und <irq> sind nur für exotische Adressen und Interrupts notwendig.

**SET KISS1=<com Nr.>,<Adresse>,<irq>**

Es sind nun 4 KISS-Schnittstellen mit KISS1..4=1..4 aktivierbar. Wie auch bei der TOKENCOM-Schnittstelle sind auch hier die Einstellungen der <Adresse> und des <irq> nur bei Abweichungen vom Standard notwendig. Wird die KISS-Schnittstelle nicht benötigt, kann sie mit SET KISS1=0 deaktiviert werden.

**B-LOG.EXE:**

Auf der Diskette ist desweiteren ein Programm B-LOG.EXE. Es legt eine Datei LOG.DAT an. Wird B-LOG.EXE von der AUTOEXEC.BAT vor dem Programmstart TNN.EXE aufgerufen, so bekommt man eine lückenlose Übersicht über die Bootvorgänge des Knotenrechners.

**TNN176.PAS**

Durch die externen Kommandos wurden zusätzliche Einträge in der PAS-Datei erforderlich. Um die Lesbarkeit der PAS zu verbessern, werden von TNN automatisch Kommentare hineingeschrieben.

Die Datei hat jetzt folgenden Aufbau und MUSS von Sysop entsprechend editiert werden:

Beim Node-Ident ist zu beachten: Werden klein geschriebene Zeichen oder andere Sonderzeichen verwendet, so kann der Knoten **N I C H T** mit dem Node Ident oder ALIAS von dem / den Userzugängen connected werden. Also „c Lippe“ oder „c Lippe-1 ... -15“ werden verhindert. Wer es zulassen möchte **M U S S** den ALIAS komplett groß schreiben.

```
; TheNetNode Konfiguration File
;
; DO NOT CHANGE THE ORDER OF THE KONFIGURATION LINES !
; DO NOT CLEAR ANY LINES !
;
; NET/ROM-Sysop-Password, 80 Characters (01234567890123...)
12345678901234567890123456734567890 usw. bis 80 Zeichen...
;
; Console Password (Geheim)
Geheim
;
; Node Ident (Test)
Test
;
; Node MyCall (XX0XX)
XX0XX
;
; Workpath, Path to the Help-Files (\TNN\)
; TNN should be started from this path.
C:\TNN\
;
; Path to the executable Text-Files (\TNN\TEXTCMD\)
C:\TNN\TEXTCMD\
;
; Path to the extern Programs for User (\TNN\USEREXE\)
C:\TNN\USEREXE\
;
; Path to the extern Programs only for Sysop (\TNN\SYSEXE\)
C:\TNN\SYSEXE\
;
; Path to the PACSAT-Files (\TNN\PACSAT\)
C:\TNN\PACSAT\
```

Verzeichnisstruktur:



**STARTPATH**

Hier stehen das ausführbare TNNxxx.EXE und die Konfigurationsdateien (TNNxxx.PAS TNNxxx.STA MEARD.TAB TNN176.TNB). Dieser Pfad ist nicht einstellbar. Er richtet sich immer nach dem Ort, an dem TNNxxx.EXE steht. Wird TNN mit „ A:\TNNxxx.EXE „ gestartet, so werden im gleichen Verzeichnis auch die Konfigurationsdateien gesucht und geschrieben.

**WORKPATH**

enthält alle von TNN verwendeten Dateien ( .TNB, .XHF), .TXT, sofern es sich um Sytemmeldungen handelt (HELP.TXT, SUSPEND.TXT, QUIT.TXT, CTEXT.\*).

TNN wechselt nach dem Start automatisch auf das unter WORKPATH eingetragene Verzeichnis.

**TEXTCMD**

Hier gehören die knotenspezifischen Texte hin, die unter HELP.TXT angegeben sind. (MAP.TXT, INFO.TXT, AKTUELL.TXT usw.)

**USEREXE**

Hier stehen die externen Kommandos, die von allen Usern ausgeführt werden dürfen.

**SYSEXE**

Hier stehen alle externen Kommandos, die nur mit SYSOP-Privileg ausgeführt werden dürfen.

**PACSAT**

Unterverzeichnis für BROADCAST-Betrieb. Siehe hierzu **ANHANG E**:

Folgende Verzeichnisstruktur arbeitet problemlos für Systeme mit Harddisk und mit RAMDISK:

```
C:\TNN\  
C:\TNN\MSG\  
C:\TNN\TEXTCMD\  
C:\TNN\USEREXE\  
C:\TNN\SYSEXE\  
C:\TNN\PACSAT\
```

Es stehen TNNDOS.EXE sowie TNNGNU.EXE zur Verfügung.

Die Bearbeitungsreihenfolge der Kommandos ist: INTERN - TEXTCMD - USEREXE - SYSEXE.

TNN176.TNB

Mit der TNN176.TNB wird die Grundeinstellung des Knotens vorgenommen.

```
;
outp 0 1      ; Grundeinstellung der seriellen Schnittstelle
....
outp 7 1
;
esc i DG9FU-13 EAMHOST  ; HOST Call der ist wenn y=n gesetzt ist connectbar.
;
; Host-IF Y=0 nicht zu connecten...oder Anzahl der feigegebenen Kanale 1..30
esc y 0
;
; Tokenring Geschwindigkeit
esc t 19200
;
; diverse L2 und L3-L4 Parameter
;
par 1 20 ; NoAck Buffer
par 2 3600 ; User Timeout (s)
par 3 3 ; Config speichern in (10min)
par 4 4800 ; DAMA-Speedfaktor
par 5 10 ; DAMA-MaxPri cnt (Maximaler Zählerstand)
par 6 5 ; Maximale Verstöße gegen DAMA Protokoll dann Disconnect.
par 7 50 ; DAMA Tout Wartezeit in ms wenn ein User nicht antwortet.
par 8 0 ; command.log 0=kein 1=Sysop.. 2=Alle Befehle festhalten
par 9 1 ; sysop.log 0=kein 1=wird geführt
par 10 2 ; Downport für Calls die nicht in der MH Liste stehen
par 11 15 ; TestSSID verwendete SSID für L+ Links
par 12 1 ; ConvSSID Verwendete SSID für Convers
;
;-----
;
; Mögliche Linkeinträge (früher Local-Einträge)
; LINK +/- <Typ> <Port nr> <Alias> <Call> [<Digi> <Digi>]
;                               [Wahlweise 1-2 Digi's]
;                               Link Call
;                               Alias 6 Zeichen
;
;      L  = Local keine Messung
;      L+ = Localer Link wird gemessen
;      F  = FlexNet
;      N+ = INP
;      N  = NetRom
;      N- = Alte TheNet
;
li + L+ 5 KSBOX DB0EAM-3
li + L 3 LOH DB0VFK
li + N 6 GOE DB0GOE
li + F 9 BIDFLX DB0BID
li + N 9 BIDTNN DB0BID-7
li + N 9 HR DB0KH DB0BID
li + N 11 HR DB0KH
;
; Locale Mailbox kann nun unabhängig von einem "eigenen" Port gesetzt
; werden. m call setzt die Mailbox M - löscht den Eintrag
m DB0EAM-3
;
;-----
```

```

;
; PORT <NR> [<CMD>[=<PAR>]] [<CMD>[=<PAR>]] [...]
;
;
; CMD:
; Off          Port abschalten
; ON           Port einschalten (nur Linux),
; SCC          Port auf USCC-Karte einstellen (DOS/GNU) bzw.
;              keine TNC-Parameter setzen (nur Linux)
; Tokenring    Port auf Tokenring schalten
; Vanessa      Port auf Vanessa schalten
; EXtern       Port auf den extern geladenen Treiber schalten
; KISSl..4     Port auf Kiss ohne CRC
; RKissl..4    Port auf Kiss mit RMNC-CRC
; SMACKl..4    Port auf Kiss mit SMACK-CRC
; MODEMl..3    Port auf Modem (TNC3)
; HSBUS        Port auf HSBUS (TNC3)
; LOOP         Port auf Loopback (NICHT bei Linux)
; Name         Port-Namen vergeben, z.B. Name=User1k2
; MODE         Port-Mode setzen, Format: 9600d
;              Baudrate: 300...4915200
;
;              Flags:
;              d : Duplex
;              c : DCD bei 1k2-Modem
;              r : ext. Takt (rx)
;              t : ext. Takt (tx)
;              e : ext. Takt beide (Vanessa)
;              m : Multibaud-Kopplung (Vanessa, SCC)
;              z : NRZ statt NRZI
; MAXFRAME <wert> setzen
; TXDelay <wert> setzen
; DAMA on/off .. Port auf DAMA Master setzen
; CTEXT on/off .. Ctext senden
; SYSOP on/off .. nach Connect nur SYS möglich, gilt nur
;                für Verbindungen zum CCP, NETROM und
;                TCPIP gehen weiterhin
; MH on/off .. MH-Liste führen
;
;-----Port 0 setzen:-----
port 0 NAME=70cm_1200;
port 0 VANESSA;
port 0 Mode=1200m;
port 0 Maxframe=2;
port 0 TXdelay=20;
port 0 CTEXT=on;
port 0 DAMA=on;
port 0 SYSOP=off;
port 0 MH=on;
be 0 10 0 ID1200;
be 0 = 1200 Bit/s User-QRG TheNetNode Digi DB0EAM : Rx =438,400 MHz -7,6 MHz
;-----Port 1 setzen:-----
;-----Port 2 setzen:-----
usw. die anderen Ports ....
;-----
;
; MH-Liste ; Größe der MH-Liste (Max 5000)
MH = 500
; L3MH-Liste
L3MH = 500
;
; Convers-Einstellungen
conv c DB0GOE; ; Goettingen
conv c DB0II 9 DB0BID; ; Mönchengladbach L2
;

```

```
; Prompt-Einstellungen
;          %a = Alias des Knoten          %c = Ident User
;          %d = Ident des Knotens         %C = Ident User mit SSID
;          %D = Ident des Knotens mit SSDI %r = ein Carriage Return
;          %t = Uhrzeit des Knotens HH:MM %0 = Keine Aussendung
;
pr =%C de %d (%t) >%r
;   ^^ wichtig: kein Leerzeichen hinter dem „=“
;
; sonstige Funktionen
; SUS + 0   DB5xxx      ; Sperrungen Port 0
; SUS + 254 DB5xxx      ; Sperrungen Level 2
; SUS + 255 DB5xxx 2    ; Beschränkung auf 2 ports
; SUS + 0    DB5xxx 0    ; Sperrungen von Usern
;
sus + 254 DG7xx
sus + 255 DG7xx 0
;
```

**SYSOP-Befehle:**

Die folgenden Befehle sollten nun den Sysop in die Lage versetzen, den Knoten für die örtlichen Belange einzurichten.

(BE)ACON <port> = <text>

Setzt bzw. ändert die Bakentexte für die einzelnen Ports.

(BE)ACON <port> =

Entfernt den Bakentext.

(BE)ACON <port> <mins> <metric> <call oder alias via call>

Schaltet die Bake ein.

Port 0..16	Portnummer.	
mins 0, n	0 =	Bake aus.
	n =	Intervall in Minuten.
		Default ist n = 10 Minuten.
metric 0, 1, 2, 3	0 =	Keine Metric.
	1 =	Metric.
	2 =	Zusätzlich STAT.
	3 =	Datenbank gerechte Ausgabe zum direkten Einlesen.
call oder alias	QST =	UI-Frame von Knoten an Call 'QST'.

Es gibt Terminalprogramme, welche die METRIC- und/oder STAT- Bake direkt auswerten und anzeigen können. Besonders sind an dieser Stelle zu erwähnen: THP (Turbo Host Packet) und TOP (The Other Packet).

Am Knotenterminal kann diese Bake auch mitgeschrieben werden. Eingabe: <ESC> MU + METRIC STAT oder <ESC> MCU + METRIC STAT. Beim Neustart wird der Monitor ausgeschaltet.

Die Bake mit metric=1 hat diese Informationen:

980208 210306 Up= 101148 Mem=1773568 Buf=5956 Rps= 524
Lnk=154 Cir= 45 Sum=2602324027 Thr= 51464
980208 210406 Up= 101159 Mem=1773568 Buf=6645 Rps= 522
Lnk=154 Cir= 43 Sum=2602669277 Thr= 48512

980208 210306	Aktuelles Datum und Uhrzeit.
Up=101158	Betriebszeit in Tagen, Stunden und Minuten. 10 Tage, 11 Stunden und 58 Minuten.
Mem=1773568	Freier DOS RAM (Coreleft, Bytes)
Buf=5956	Freier TNN-Buffer (Blöcke).
Rps=524	Hauptschleifendurchläufe/Sekunde.
R=0	Anzahl der Token-Recoveries.
Lnk=154	Anzahl der aktiven L2-Links.
Cir=45	Anzahl der aktiven Circuits (L4-Links).
Sum=229961	Zählerstand des Statistik-Gesamtzählers.
Thr=51464	Derzeitiger Datendurchsatz in Baud.

Bei metric=2 wird zusätzlich bei jedem empfangenen L3-RTT Meßframe ein UI-Paket des Knotens an METRIC abgestrahlt:

```
10.12.97 16:26:52 DB0LBG-7(09)
L3RTT=15490ms L3SRTT=7350ms (7350ms/6760ms) L2SRTT=820ms SUM=4302022
10.12.97 16:26:54 DB0NHM(10)
L3RTT=840ms L3SRTT=790ms (1090ms/500ms) L2SRTT=640ms SUM=1750605
10.12.97 16:27:02 DB0BID(09)
L3RTT=2210ms L3SRTT=1340ms (1780ms/900ms) L2SRTT=680ms SUM=1510236
```

Aktuelles Datum und Uhrzeit

DB0LBG-7

(09)

L3rtt=000185ms

L3SRTT=1. Zeit (2. Zeit / 3. Zeit).

1. Zeit

2. Zeit

3. Zeit

Rufzeichen des Nachbarknotens.

Empfangsport des Meßframe.

Soeben gemessener Level-3 RTT.

Ermittelter SRTT aus der Messung.

Bei FlexNet mittel aus 2. Zeit und 3. Zeit.

Von diesem Knoten ermittelter SRTT in ms.

ohne Erklärungen ausgestrahlt.

Datum	Uhrzeit	Betriebs Zeit (s)	Freies Ram	Freie Blöcke	RPS	LNK	CIR	Summe Byte	Durch- satz
13.03.98	09:12:06	41159	1642496	7122	696	128	32	114642456	30488
13.03.98	09:13:06	41219	1642496	7201	689	129	33	114924309	34568
13.03.98	09:14:06	41279	1642496	7409	695	126	32	115162851	32848
13.03.98	09:15:06	41339	1642496	7276	674	130	35	115401514	32264
13.03.98	09:16:06	41399	1642496	7296	707	121	33	115613708	30120

Call	Port	Datum	Uhrzeit	L3RTT	L3SRTT mittel	L3SRTT MyQual	L3SRTT HisQual	L2SRTT	Übertragene Byte
DB0BID	9	10.12.97	16:02:44	330	1190	1590	800	590	1488645
HB9AK	9	10.12.97	16:03:11	14350	6320	6320	8220	350	2026217
DB0GOE	6	10.12.97	16:03:16	480	620	620	700	420	29427844
DB0BRO	10	10.12.97	16:03:26	3350	2620	2620	2310	860	4626794
DB0LIP	8	10.12.97	16:03:45	810	1230	1230	610	900	13049378
DB0KH	11	10.12.97	16:04:30	490	1210	1210	1130	420	24740302

(BE)ACON <port> 0 0 ID

Schaltet die Bake wieder aus.

(CL)EAR

Löscht alle Statistikeinträge.

(CONV)ers (C)stat

...oder im Conversmode mit: /Link

Zeigt die Liste und den Status der Nachbar-Convers-Host an sowie die Laufzeiten zu allen Convers-Host und deren Versionsnummer. **NUR** im Sysopmodus wird zusätzlich der eingetragene Weg (ob l4 oder l2 mit Port und Via-Einträgen) angezeigt.

(CONV)ers (C)stat <call>

...oder im Conversmode mit: /Link <call>

Nimmt einen Nachbarknoten für Convers-Betrieb auf. Unter <call> ist das Rufzeichen des Nachbarknotens einzusetzen. Der Nachbarknoten muß dann in der Nodesliste stehen.

(CONV)ers (C)stat <call> <port> <via-call>

...oder im Conversmode mit: /Link <call> <port> <via-call>

Nimmt einen Nachbarknoten für Convers-Betrieb über eine Level2 Verbindung auf. Unter <call> ist das Rufzeichen des Nachbarknotens einzusetzen. Unter <port> ist die Portnummer anzugeben und bei Bedarf ein oder mehrere via-call.

(CONV)ers (C)stat <call> 254

...oder im Conversmode mit: /Link <call> 254

Gibt einen Nachbarknoten für Convers-Betrieb frei. Unter <call> ist das Rufzeichen des Nachbarknotens einzusetzen. Es wird zu dem <call> KEIN Link aufgebaut sondern TNN erwartet, daß dieses <call> den Link aufbaut. Dieses ist z.B. für Wampe Rechner die zwar selbst einen Link aufbauen können aber nicht mit einem ankommenden Connect zurecht kommen. Der Eintrag wird gekennzeichnet mit: (trusted host).

ES MÜSSEN UNBEDINGT LOOPS VERHINDERT WERDEN. DESHALB: BITTE ABSPRECHEN !

(CONV)ers (C)stat - <call>  
...oder im Conversmode mit: /Link - <call>  
Löscht den Eintrag.

(CONV)ers - intern -

**PingPong Convers aus Sysopsicht:**

#### Links:

Links werden mit dem „/links“ Befehl eingetragen, der auch außerhalb des Convers mittels „conv c“ verfügbar ist. So eingetragene Links sind permanent und werden in bestimmten Intervallen (9s,150s,300s,600s,...,3h) connected. Es gibt auch nichtpermanente Links, diese Option ist aber durch Konfiguration einstellbar. Diese Links werden bei einem Linkabriß automatisch ausgetragen, der Knoten connected den neuen Partnern nicht hinterher. (Das Eintragen geschieht durch Eingeben des „/.host“ Befehls.)

Syntax ist: /l [-][call [port [via]]]  
bzw. außerhalb von Convers: conv c [-][call [port [via]]]

- Dient zum Löschen eines Eintrags (<call> reicht aus).  
<call> Ist das Rufzeichen des Nachbarknotens.  
<port> Downlinkport (wird nur bei L2 Verbindungen gebraucht).  
<via> Via-call (optional für L2 Verbindungen).

Für L4-Verbindungen genügt die Angabe des Calls des gewünschten Knotens, es wird NICHT versucht, eine L2-Verbindung bei fehlendem Nodeeintrag aufzubauen. Dem Sysop wird bei diesem Befehl unter jedem Linkeintrag der Connectweg in Klammern angezeigt, z.B.:

im Falle L4 (DB0AGM),  
im Falle L2 (DB0XYZ on port 2 via DB0ZXY).

#### Hilfe:

Ist in einer Datei namens **CONVERSD.XHF**. Diese enthält alle Texte, getrennt durch Markierungen. Es ist erlaubt, die Texte zu ändern, Reihenfolge spielt bis auf den obersten Abschnitt keine Rolle. Die Markierungen sind durch Convers vorgegeben, Einfügen von neuen Markierungen ist daher sinnlos. Die Datei enthält Umlaute nach ISO, bitte beim Überarbeiten daran denken. Wird die Hilfe angefordert, konvertiert der Convers die Umlaute selber in die vom User gewünschte Form.

#### Sonstige Dateien:

**CONVERS.PRS** enthält die persönlichen Daten aller, die diese am eigenen Knoten eingegeben haben. Diese Datei ist ebenfalls im Textverzeichnis, wird aber beim Beenden von TNN im Configverzeichnis gesichert bzw. beim periodischen Sichern der Konfiguration. Darunter fallen der Personaltext, die Zeichensatzwahl und die Zeilenbreite.

#### Loop detected:

Loops werden nun überwacht und der entsprechende Link für eine Stunde „aus dem Verkehr gezogen“. Das Auftreten und die Häufigkeit wird in der **CONV-C-Tabelle** angezeigt.

#### (DCD)

Das DCD- Kommando zeigt den in der Software erkannten DCD Zustand an.

P00	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12	P13	P14	P15
t		rt	rt	rT	T			OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

OFF = Port nicht in Betrieb,  
r = DCD an,  
R = DCD an und noch Frames im Empfänger in der Vanessa,  
t = PTT an,  
T = PTT an, noch Frames zu senden in der Vanessa.

#### (DOS) <kommando>

Zugriff auf COMMAND.COM. Hiermit lassen sich viele Operationen durchführen. Natürlich muß hierzu COMMAND.COM auch den Zugriff auf die entsprechenden Erweiterungsprogramme haben. Doch Vorsicht ist unbedingt angesagt! Bei einem Fehler habe ich zumindest 50 km Autofahrt und evtl. einen kräftigen Fußmarsch gewonnen. Also interaktive Eingaben vermeiden! Außerdem sollte man nur Programme starten, die sehr schnell wieder beendet sind, weil während der Ausführung des Programms der gesamte Knoten angehalten wird. Programme, die länger als 1 Minute laufen, führen zu einem Rechnerneustart (nicht bei LINUX).

Beispiel:       DOS DIR       listet das aktuelle Verzeichnis.  
              DOS DIR A:   listet LW A.

Bei LINUX heißt das Kommando SHELL, und beim TNC3 heißt das Kommando TOS.

**(DX)CUSTER <call>**

setzt das Call des lokalen Cluster. Das Cluster muß nun nicht mehr Bestandteil des eigenen Knotens sein, sondern es kann auch ein beliebiges Cluster eingetragen werden.

**(DX)CLUSTER -**

löscht das gesetzte Call.

**(E)DIT <filename.ext>**

Legt den hinter EDIT angegebenen File-Name im Workpath an. Mit (E)dit kann jedoch kein bestehendes File zum Ändern aufgerufen werden. Bestehende Files werden ersetzt!.

**(E)DIT <path\filename.ext>**

Auch die Angabe eines Pfades ist möglich.

Ist der Editiermodi bereits durch einen anderen Sysop aufgerufen, so bekommt man die Meldung :

Edit-Mode in use by other Sysop

**(K)ill <port> <msg>**

Unterbricht alle Level-2 Verbindungen auf dem <port> nach Sendung der <msg>.

**(K)ill <call> <msg>**

Unterbricht alle Level-2 Verbindungen zu dem <call> nach Sendung der <msg>.

**(K)ill \* <msg>**

Trennt alle Level 2 QSO auf dem Digipeater nach Sendung der <msg>.

**(L3)MHEARD -**

Setzt die Zählerstände der Level 3-Statistik auf 0, aber löscht nicht die L3MH-Liste.

**(L3)MHEARD = <anzahl>**

Setzt die L3MH-Liste auf die <anzahl> Rufzeichen. Man bedenke aber, daß sie auch gespeichert werden müssen. Löschen der MH-Liste geschieht immer dann, wenn ihre Größe verringert wird. Damit sind dann aber auch die L3- Daten verloren.



**(L)INKS** +/- <typ> <port> <alias:call[-\*]> [<digi1> <digi2>]

Nimmt <typ> mit <alias:call> über <port> in die Linkliste auf. Für die Anbindung von Digipeatern ohne TheNetNode-Protokoll. Die Digipeater <digi1> und <digi2> können mit angegeben werden. Es werden nur Nodes angenommen die, unter Links eingetragen sind. Hierbei wird auch der eingetragene Port beachtet.

+/- + Eintragen und - Austragen des Links.

Typen:

**L** = Local  
Call und Alias wird mit einer Laufzeit von 4000 ms in die NODES-Liste übernommen, es wird aber keine Laufzeitüberprüfung durchgeführt. In der Routesliste wird er ohne Eintrag in Status geführt.

**L+** = Local  
Wie oben, jedoch wird die Erreichbarkeit geprüft und die Laufzeit gemessen. Kann die Laufzeit ermittelt werden, so wird dieses Ziel zur besseren Übersicht als conn. im Status geführt.

**F** = FlexNet  
Port arbeitet mit FlexNet Protokoll.

**N** = NetRom  
Aktueller TheNet-Nachbar. Übernahme in Nodesliste mit der ermittelten Laufzeit.

**I** = InterNode Protokoll  
Wird jedoch in die Linkliste als „N“ eingetragen und automatisch auf „I“ geändert.

**#ALIAS**  
ALIAS - Einträge die mit einem „#“ davor gekennzeichnet sind, werden zwar in die örtliche Nodesliste aufgenommen aber nicht als NODE weiter verteilt.

**(LOA)D** <filename.ext> oder **(LOA)D** <path>\<filename.ext>

Lädt <filename.ext> auf die Diskette oder Festplatte beim Knoten. Der <filename.ext> darf maximal 8 Zeichen + 3 ext. lang sein und muß folgendes Format haben:

LOAD TNN176.EXE

**Vorsicht** bei der Auswahl des Filenamens, damit nicht das bisherige TNN-Programm versehentlich überschrieben wird. Gegebenenfalls einen noch nicht verwendeten Filenamens verwenden und nach dem Umkopieren in das gewünschte Verzeichnis mit DOS REN ... umbenennen.

Nach der Eingabe von „LOAD <filename.ext>“ wartet der Rechner auf die Zeichenfolge #BIN#Länge (z.B. #BIN#82345). Die Zahl hinter #BIN# steht für die Programmlänge in Bytes (dezimal). Auf diese Zeichenfolge wird mit einem #OK# reagiert. Nachdem der SYSOP dieses #OK# empfangen hat, kann mit der Übertragung begonnen werden.

Die oben beschriebene Prozedur entspricht derjenigen, die von THP, SP oder GIPSY als AUTOBIN-THP bekannt ist. Zum Einladen eignen sich diese Programme also ganz besonders.

Nach erfolgreichem Einladen gibt der Knoten eine Checksumme (CRC) aus. Diese Checksumme MUSS derjenigen entsprechen, die auch von den o.a. Programmen ermittelt wird. TNN wurde daran angepaßt.

Siehe hierzu auch noch die Möglichkeiten die STARTCNT.EXE ermöglicht.

**(M)AILBOX** <call>

setzt das Call der lokalen Mailbox. Die Mailbox muß nun nicht mehr Bestandteil des eigenen Knotens sein, sondern es kann auch eine beliebige Mailbox eingetragen werden.

**(M)AILBOX** -

löscht das gesetzte Call.

**Mailbake** ist kein Befehl in TNN sondern eine Funktion. UI-Frames können auf einem bestimmten Port gesendet werden. Dazu wird der Knoten - Alias mit SSID verwendet. Der SSID gibt hierbei den Port an, wo die Bake gesendet werden soll.

DB0GSO-10 to MAILS via GSO-4

< 15:36:57 > DPBOX v5.07.00

Mail for : DB6KF DC8KN DG1KMN DG3KMB DK3BR DL0GSO

sende die Mailbake der DP-BOX auf Port 4 (GSO der Alias, -4 der SSID = Port). Über diesen Weg könnte z.B. auch eine DP-BOX einen PACSAT Broadcast auf einem bestimmten Port abstrahlen.

**(MH)EARD -**

Setzt die Zählerstände der USER-Statistik auf 0, aber löscht nicht die MH-Liste, die bei den Userports benötigt wird.

**(MH)EARD = <anzahl>**

Setzt die MH-Liste auf die <anzahl> Rufzeichen. Man bedenke aber, daß sie auch gespeichert werden müssen. Die Längenänderung führt ggf. zum Verlust der am Ende der Liste stehenden Calls. Die Anzahl ist auf 5000 begrenzt.

**(P)ARMS <nummer>**

Zeigt den eingestellten Wert dieses Parameters sowie den einstellbaren Bereich an.

**(P)ARMS <nummer> <wert>**

Beispiel: p 1 20

Setzt Parameter 1 (NoAckBuf) auf 20. Die Änderung wird quittiert mit:

1: NoAckBuf = 20 (7...127)

**(PAC)SAT <parameter>**

zeigt im SYSOP-Mode die eingestellten Parameter an.

BROADCAST-Parms:				
01:Timer	200	02:Frames	15	03:Diskfree 1000

**(PAC)SAT <parameter> <nummer> <wert>**

Beispiel: pac p 1 300

Setzt Parameter 1 (PacSatTimer) auf 300. Die Änderung wird quittiert mit:

1: Timer = 300 (0...10000)

Die Parameter in einzelnen:

01: PacSatTimer : 200  
 02: PacSatFrms : 15  
 03: PacMaxMail : 1000

Mit PAC P 01 wird eingestellt, in welchen Intervallen der Knoten die PacSat-Frames sendet, dieser Parameter muß so eingestellt werden, daß der TNC nicht überläuft. Mit PAC P 02 wird die Anzahl der Pakete die, gleichzeitig gesendet werden, eingestellt. Das sollte bei 9k6 20 oder so sein, (pi \* Daumen 5 Frames pro Sekunde), sonst lohnt sich das nicht. Mit P 03 kann die max. Anzahl der Mail im PACSAT-Server eingestellt werden. Sind beim S&F mehr als mit PAC P 03 angegebenen Mail im Server, so werden die ältesten Mail gelöscht.

Über die Timersteuerung kann eine optimale Kanalauslastung erreicht werden, die von vereinzelt Aussendungen auf einem User-Einstieg bis zur Dauertastung alles ermöglicht, dabei werden zwei Taktiken unterstützt:

- nur PacSat-Ausgänge:  
 Dauertastung, für den ISM-Bereich. TNN macht dann 24-Stunden am Tag 365 Tage im Jahr Krach mit Dauerträger. Dazu ist auf dem Port, auf dem PacSat betrieben wird, ein DAMA-Eprom zu verwenden, DAMA für den PORT aber ABZUSCHALTEN. Die DAMA-Bestätigung wird benutzt, um die Dauertastung zu gewährleisten. P 02 bestimmt, wieviele Frames auf einmal zum TNC geschickt werden, der muß immer einen ausreichenden Vorrat haben, damit er die PTT nicht losläßt, das darf aber auch nicht zu viel sein, sonst läuft er über. Je nach Ringbelastung muß dieser Wert so sein, daß der TNC auch ohne, daß er Daten kriegt, eine Weile durchhält. Bei 19k2 Tokenring habe ich den Param p 02 auf 14, bei 38k4 kann das weniger sein, bei sehr stark belastetem Tokenring mehr. P 01 gibt an, wie lange gewartet wird, wenn KEINE DAMA-Bestätigung kommt, dieser Parameter sollte auf 1000 stehen, ist aber eigentlich unerheblich (solange die Bestätigungen kommen).
- Gemischte Ein/Ausgänge:  
 Hier muß man entweder DAMA für den Port einschalten und ein DAMA-Eprom verwenden oder KEIN DAMA-Eprom verwenden und kein DAMA machen, DAMA-Eprom und kein DAMA geht NICHT (dann macht der Knoten Dauertastung)

Dauertastungs-Ausgänge sind unbedingt vorzuziehen, sie setzen 82MB Daten mit 9k6 am Tag um, das ist das 10-fache eines normalen 9k6-Einstiegs. Außerdem hat jeder User was davon, nicht nur Dauersauger XYZ. Ein Broadcast-Ausgang mit 1k2 ist Unsinn und sollte nicht betrieben werden. Es muß aber unbedingt darauf geachtet werden, daß der Sender auch für Dauertastung geeignet ist. Der Tnet-Mini ist z.B. im Original nicht geeignet!

- Neben BOX kann man jetzt auch mit C PACSATCALL in die Box!

**(PAC)SAT c <call>**

Setzt das Rufzeichen des BROADCAST-Server.

**(PO)RT <Port Nr.> <Befehl 1> <Befehl 2> .....**

Dient zum einstellen der Port-Parameter und der Schnittstellen.

Als Port-Befehle sind gültig:

Off	Port abschalten
ON	Port einschalten (nur Linux),
SCC	Port auf USCC-Karte einstellen (DOS/GNU) bzw. keine TNC-Parameter setzen (nur Linux)
Tokenring	Port auf Tokenring schalten
Vanessa	Port auf Vanessa schalten
EXTErn	Port auf den extern geladenen Treiber schalten
KISS1..4	Port auf Kiss ohne CRC
RKiss1..4	Port auf Kiss mit RMNC-CRC
SMACK1..4	Port auf Kiss mit SMACK-CRC
MODEM1..3	Port auf Modem (TNC3)
HSBUS	Port auf HSBUS (TNC3)
LOOP	Port auf Loopback (NICHT bei Linux)
Name	Port-Namen vergeben, z.B. Name=User1k2
MODE	Port-Mode setzen, Format: 9600dz Baudrate : (300...4915200) Flags : (c...z). Zugelassene Flags sind: c : DCD bei 1k2-Modem d : Duplex r : ext. Takt (rx) t : ext. Takt (tx) e : ext. Takt beide (Vanessa) m : Multibaud-Kopplung (Vanessa, SCC) z : NRZ statt NRZI
MAXFRAME <wert>	setzen
TXDelay <wert>	setzen
DAMA on/off	Port auf DAMA Master setzen
CTEXT on/off	Ctext senden
SYSOP on/off	nach Connect nur SYS möglich, gilt nur für Verbindungen zum CCP, NETROM und TCPIP gehen weiterhin
MH on/off	MH-Liste führen

**(OF)f :** Port ausgeschaltet.

**(ON):** Nur bei Linux: Port eingeschaltet. Da im File tnn.ini eine feste Zuordnung zwischen den Ports und der Hardware sowie des KISS-Modus eingestellt wird, ist es nicht erforderlich, die Hardware einzustellen. Der Parameter ON ist daher der sicherste, aber KISS ist auch immer gültig zum einschalten.

**(TO)kenring :** Frames an diesen Port werden auf die Tokenring-Schnittstelle gelegt. Die Tokenring-Schnittstelle liegt default auf COM 1, wenn sie nicht in der START.BAT anders definiert wird. Bei Linux wird die Bezeichnung TOKENRING nur akzeptiert, wenn auch in tnn.ini der Port als Tokenring-Port deklariert ist.

...(MO)de: Keine Auswirkung auf den Tokenring, aber auf die L2-Timer !

**(VA)nessa :** TNN ist sowohl für den Tokenring als auch für den gleichzeitigen Einsatz von Vanessakarten geeignet. Für Frames an diesen Port wird, über den in der TNN-Soft implementierten Vanessatreiber, die entsprechende Vanessakarte über den Rechnerbus angesprochen und unter (PO)rts wird der entsprechende Port mit „Vanessa“ gekennzeichnet (Jedoch NUR, wenn auch eine VANESSA eingebaut ist). Bei Linux wird die Bezeichnung VANESSA nur akzeptiert, wenn auch in tnn.ini der Port als Vanessa-Port deklariert ist.

...(MO)de: <speed>d :	Vollduplex. Bei der Vanessakarte bleibt der TX nach der letzten Sendung noch für ca. 1 Minute getastet. Weitere Frames werden ohne TXDelay gesendet. Beim TNC im Tokenring ist diese Zeit im Eprom festlegbar.
...(MO)de: <speed>e :	Externer Takt.
...(MO)de: <speed>m :	Dual - Speed - Port (Nur mit und auf den beiden Ports der entsprechenden Vanessakarte möglich).
(SCC) :	Die BayCom USCC-Karte wird bei der DOS- bzw. GNU-Version nun intern unterstützt! Bei Linux bedeutet der Parameter SCC, daß für diesen Port keine Initialisierung der L1-Parameter erfolgt (TXD, Persistence, Slottime, etc.). Dies ist Aufgabe eines Initialisierungsprogramms, das vor TNN aufgerufen werden muß - alternativ gelten die TNC Default-Werte. Der Parameter SCC ist bei Linux nur wirksam, wenn der Port in tnn.ini entsprechend eingetragen ist. Die Mode-Werte werden bei Linux ignoriert.
...(MO)de: <speed>c :	Software DCD für AFSK-Modems.
...(MO)de: <speed>e :	Externer Takt. (Für DF9IC-Modem).
...(MO)de: <speed>z :	NRZ statt NRZI. (Für DF9IC-Modem).
Hinweis:	Soll DF9IC-Modem an eine Vanessa angeschlossen werden, so benötigt es NRZ-GAL ! Mit den NRZI-GAL soll der externe Clock nicht funktionieren.
(KISS1)..(KISS4) :	Die Frames an diesen Port werden über den in der TNN-Soft implementierten KISS-Treiber, an die in der START.BAT definierte COM-Schnittstelle, über den Rechnerbus geleitet. Unter (PO)rts wird der entsprechende Port mit „Kiss1 .. Kiss4“ gekennzeichnet und mit (MO)de der Speed sowie das CRC-Verfahren festgelegt. Bei Linux kann mit dem Parameter KISS jede Port-Art eingeschaltet werden. Die Zahl für den KISS-Port wird bei Linux ignoriert.
(SMACK1)..(SMACK4) :	KISSLINK mit SMACK CRC. Bei Linux wird die Bezeichnung SMACK nur akzeptiert, wenn auch in tnn.ini der Port als SMACK-Port deklariert ist. Die Zahl für den SMACK-Port wird bei Linux ignoriert.
(RKISS1)..(RKISS4) :	KISSLINK mit RMNC-CRC. Bei Linux wird die Bezeichnung RKISS. nur akzeptiert, wenn auch in tnn.ini der Port als RKISS-Port deklariert ist. Die Zahl für den Kiss-Port wird bei Linux ignoriert.
<u>Anmerkung:</u>	Bei EXTERNEN-Treibern wird der TYP der Hardware direkt vorgenommen und ist nach dem Starten der Software bereits unter dem jeweiligen Port sichtbar.
DAMA on / off	Port auf DAMA Master setzen. Bei der Linux-Version funktioniert DAMA NUR mit Tokenring oder Vanessa !  (ACHTUNG DAMA-BIT im EPROM der Tokenring-Software brennen!)
CTEXT on / off	Ist CTEXT on wird generell der CTEXT.TXT gesendet. Ist noch zusätzlich ein CTEXT.n (n=Portnr.) vorhanden, so wird auch dieser auf den entsprechenden Port gesendet. Ist CTEXT=0, so werden auch evtl. vorhandene <call>.MSG Files nicht gesendet.
SYSOP on / off	Setzt den angegebenen Port auf SYSOP-Mode. Diese Funktion soll dem Sysop helfen, seine Wartungsarbeiten durchführen zu können. Der Knoten nimmt ohne eine SYSOP-Privilegierung auf diesem Port keine Kommandos mehr an. Die Umstellung kann auch während des Betriebes erfolgen, ohne daß die User disconnected werden. NetRom und TCP/IP gehen weiterhin.

Semiduplex Links können auch auf SYSOP gesetzt werden. Dadurch wird ein unerlaubtes „Einsteigen“ auf den Links verhindert. Der Level 3-4 Link zum Nachbarn bleibt unberührt.

**MH on / off**

Das Führen der MH-Liste auf den Interlinks kostet Rechnerzeit und die MH-Liste wird unübersichtlich. Sie wird geführt, um bei mehreren Userzugängen mehr Transparenz über die Zugänge zu erhalten UND damit der KNOTEN weiß, auf welchem Port er einen bestimmten User connecten muß.

**(NA)me :**

Gibt diesem Port eine spezielle Bezeichnung. Sie darf maximal 10 Zeichen lang sein und dient der Unterscheidung der Ports. Wenn z.B. mehrere Userzugänge oder Baudraten benutzt werden, ist in der MH-Liste ersichtlich, welches Call auf welchem Zugang QRV ist. Weiterhin wird der Portname benutzt, um auf einem bestimmten Port einen Connect auszusenden, unabhängig vom Eintrag in der MH-Liste. DB0EAM hat einen „70cm\_1200“ und einen „70cm\_9600“ Zugang sowie nun auf 23cm einen „23cm\_9600“. Die Port-Namen sind auch so eingestellt. Mit „c DB0XYZ 70cm\_9600“ kann ein Connect auf dem 9600\_Bit/s Port ausgesendet werden. Wird kein Port-Name beim Connect eingesetzt, so wird Port 0 benutzt, ist das Call in der MH-Liste vorhanden mit einem Eintrag auf einen anderen Port als Port 0, so wird dieser Port benutzt.

**Autoparameter:**

Einige Parameter für die Ports werden nun automatisch von TNN eingestellt, so daß nun nur noch MaxFrame und TXDelay übrig bleiben. Diese werden nun nicht mehr über die PARAMETER eingestellt sondern mit dem PORT Kommando.

$T2 = 2888 / (\text{Baudrate} / 100)$

bei DAMA wird  $T2 * 2$  genommen (Idee DG3AAH)

$IRTT = (T2 + TXDelay) * 2.$

Der IRTT wird beim Connect mit der Anzahl der noch zu digipeatenden Stationen multipliziert.

Retry = 10 bei DAMA, sonst 20.

Persistence ist bei Duplex und DAMA 255, ansonsten 255/User. Bei 0 Usern auch 255.

Es werden nur die zum Knoten connecteten User berücksichtigt. Der Knoten nimmt also nicht auf Schwarzfahrer auf der gleichen QRG Rücksicht.

Slottime = TXDelay

**(PR)OMPT**

Ohne Text zeigt die derzeitige Einstellung an:

Prompt: %c de %d (%t) >

**(PR)OMPT =<text>**

Übernimmt den String als Prompt. Der für den Prompt verwendete <text> beginnt direkt hinter dem „=,“! Enthält der eingegebene String die Zeichen %c, %d, %t, %0, so werden diese im späteren Prompt wie folgt umgewandelt:

%a	In den Alias des Knotens,
%c	In das Call des User,
%C	In das Call des User mit SSID,
%d	In das IDENT des Knotens,
%D	In das IDENT des Knotens mit SSID,
%r	In ein Return,
%t	In die momentane Uhrzeit in HH:MM,
%s	In das aktuelle Datum,
%0	Verhindert die Aussendung des Promptes.

PR=%c de %d (%t) > ergibt:

DG9FU de DB0EAM (18:30) >

Es hat sich eingebürgert, im gesamten PR-Netz UTC zu verwenden. Dies erspart einem auch das lästige Neusetzen der Sommer-/Winterzeit.

**(RE)AD**

Dient zum Lesen eines Files. Der Name muß mit Erweiterung angegeben werden. z.B.:

READ CTEXT.TXT oder READ AKTUELL.TXT.

Hier sind auch Pfadangaben erlaubt. Diese sind notwendig, wenn ein Text vom Laufwerk A: gelesen werden soll und der DOS-Pfad auf ein virtuelles Laufwerk zeigt. z.B.:

READ A:\TNN\AKTUELL.TXT.

**(READB)IN** <filename.ext>

Erlaubt das binäre Downloaden von Files. Das Verfahren ist das gleiche wie unter LOAD, nur in der Richtung vom Knoten zum Sysop.

**(RES)ET** SYSTEM

Läßt den Knotenrechner einen Kaltstart ausführen. Bei LINUX wird TNN lediglich beendet. Wird hier ein Neustart des Knotenrechners benötigt, muß man den Befehl „SHELL shutdown -r now“ verwenden.

**(RES)ET** <port>

Führt einen Reset des Port-TNC bzw. der entsprechenden Vanessakarte aus.

**(RU)NBATCH** <filename.tnb>

Führt das angegebene TNB-File aus und gibt die Bestätigung „OK“ zurück. Der Befehl RUNBATCH kann auch in einem TNB-File eingetragen werden. Er kann beliebig zwar verschachtelt werden, aber hier ist Vorsicht geboten, weil eine unendliche Rekursion nicht verhindert wird - das führt dann zur Blockade des gesamten Rechners, weil die Festplatte mit einem Temporär-File gefüllt wird.

**(S)TAT** + <call>

Nimmt <call> in die Statistik auf.

**(S)TAT** - <call>

Löscht <call> aus der Statistik.

**(SP)ARM** oder besser **Save Parameter**

Save Parameter erstellt ein File PARAMS.TNB, das alle derzeit eingestellten Parameter beinhaltet. PARAMS.TNB kann, umkopiert nach TNN176.TNB, zum Einstellen der Parameter von TNN beim Neustart verwendet werden (aber vorher auf Vollständigkeit prüfen!). Als Bestätigung kommt „PARAMS.TNB saved ...“

**(STAR)T** <programm>

Startet das angegebene Programm. Der Programmname muß vollständig angegeben werden, z.B.: Start TNNDOS.EXE. Befindet sich das Programm nicht im aktuellen Pfad, so muß dieser selbstverständlich mit angegeben werden. Der Ablauf des Startvorganges wird nun nicht mehr auf dem Bildschirm angezeigt, sondern in eine Datei STARTUP.LOG geschrieben.

**(STAR)T** <programm> funktioniert NICHT bei den DPMI-Versionen !

**(SUS)PEND** + <port> <call> <links>

Beschränkt das Rufzeichen <call> auf Port <port> auf eine Anzahl von <links>.

Beispiele:

DB2OS soll auf dem Knoten generell nur 2 Verbindungen, egal auf welchem Port, aufbauen können. Dann wird unter <port> einfach der fiktive Port 255 angegeben.

SUS + 255 DB2OS 2

Soll er nun vom USER-PORT 0 ganz ausgeschlossen werden, so ist einzugeben:

SUS + 0 DB2OS

Ein Level-2 Ausschluß erfolgt mit:

SUS + 254 DB2OS

Der Connect des Knotens ist erstmal möglich. Jedoch nach dem ersten I-Frame des Users bekommt er das File SUSPEND.TXT zugesandt und danach einen Disconnect.

**(SUS)PEND** - <port> <call>

Gibt dem Rufzeichen <call> auf Port <port> wieder das Benutzen des Knotens frei.

Ein PR-User hat sich beim BAPT darüber beklagt, daß ihm der Betreiber eines Digipeaters nur noch jeweils einen Connect mit dem Digi zur selben Zeit zur Verfügung stellt. Da dieses Problem schon häufiger für Diskussionen in PR gesorgt hat, gebe ich hiermit das Schreiben des BAPT an den Digi-Betreiber im vollen Wortlaut wieder:

Aktenzeichen 123-8 B 3581-29 vom 28.01.1997

Begrenzung auf maximal einen gleichzeitigen Connect bei dem Digipeater DB0xxx

Sehr geehrter Herr xxx,

wie mit Ihnen am 27.01.1997 telefonisch besprochen, teilen wir Ihnen hiermit unsere grundsätzlichen Aussagen mit, die von uns bezüglich einer gegen Sie vorgebrachten Beschwerde wegen der o. g. Beschränkung bei Ihrem Digipeater gegenüber dem Beschwerdeführer gemacht wurden.

Wir betrachten die o. g. Begrenzung nicht als Ausschluss im Sinne der Anlage 1, Punkt 2.4.2.7 der DV-AFuG. Ein Ausschluss auf Grund eines entsprechenden Missbrauches einer Relaisfunkstelle liegt nicht vor, vielmehr handelt es sich hier um Begrenzungen aus betrieblichen Gründen. Wir halten diese Begrenzung für eine, zum Beispiel aus Kapazitätsgründen, sinnvolle Massnahme, um möglichst vielen Funkamateuren die Gelegenheit zu geben, mit diesem Digipeater zu arbeiten. Es muß Sache des Betreibers eines Digipeaters bleiben, nach seiner Einschätzung die von ihm zur Verfügung gestellte technische Einrichtung hinsichtlich Quantität den Nutzern anzubieten. Ausserdem sehen wir keine rechtliche Grundlage, auf die sich im Amateurfunkdienst ein Anspruch auf mehrere gleichzeitige Verbindungen über eine Relaisfunkstelle begründen lässt. Für weitere Fragen stehen wir Ihnen gerne zur Verfügung.

Mit freundlichen Grüßen

Im Auftrag

gez. Jeromin (für Link)

#### (SY)SOP

Gibt eine 5er Zahlengruppe aus für das Einloggen des Sysop.

##### **PASSWORD-Eingaben:**

Die Paßworteingabe für den User wurde erweitert, so daß auch Paßworte wie bei BAYCOM-BOX erlaubt sind. Die alten 5-stelligen Antworten gehen auch. Der Trick besteht darin, das eigentliche Paßwort in einem langen String Datenmüll zu verstecken!

##### **BAYCOM-PASSWORD:**

Das wichtigste zuerst, die Methode ist kompatibel zur alten, man kann ohne weiteres einfach die 5 Antwortbuchstaben senden. Zusätzlich ist es aber möglich, die Antwort in einer Zufallszeichenkette zu verstecken. Ist die Lösung z.B. 12345, werden folgende Antworten auch als korrektes Password akzeptiert:

54321532412345221134254235244242415526

-----

12345

-----

12131415234132145132412345

-----

Da dem Lauscher nicht bekannt ist, wo die Lösung ist, kann er sich das Password nicht zusammenbasteln. Es empfehlen sich also Paßwörter mit möglichst vielen verschiedenen Zeichen, um die Verschleierung zu erhöhen.

#### (TE)ST <port>

Gibt einen 0/1 Wechsel auf dem angegebenen Port von 4 KByte Länge aus, für Abgleicharbeiten z.B. am Modem. Die Testfunktion kann nur noch der Sysop auslösen. Ruft der User die Testfunktion auf, so bekommt er ein „Port disabled“ zurück.

#### (TI)ME

Zeigt Datum und Uhrzeit des Knotens an.

Einstellen von Uhrzeit und Datum ist nur über die DOS-Befehle möglich!

Zeit mit:           **DOS TIME HH:MM**  
Datum mit:       **DOS DATE TT.MM.JJ**  
oder mit:         **DOS DATE TT.MM.JJJJ**

Doch Vorsicht! Nur „DOS TIME“ zeigt die aktuelle Uhrzeit auf dem Monitor des Knotens an und erwartet die neue Uhrzeit per Tastatur. Der Knoten steht dann bis zum RESET. Dieser erfolgt automatisch nach ca. 1 Minute durch den Software-Watchdog von TNN.

Ein wenig Mitdenken gehört auch dazu.

(TR)ace <level> <monitor>

Dieser Befehl dient zum erkennen und finden von Fehlern. Im Prinzip besteht der Trace aus 2 Funktionen. Zum einen die Ausgabe von Ereignissen und Fehlern und einem zusätzlich einschaltbaren Monitor. Es stehen folgende Module zur Verfügung. Dazu ist der Level mit einem Wert zwischen 0 (Aus) und 9 (Alles) einstellbar. Der Trace wird, wenn mehr im Monitor zu senden sind als nach Parameter NoAckBuf eingestellt ist, nicht abgebrochen sondern nur unterbrochen. D.h. es können, wenn man zu viel monitoren will, Lücken entstehen. Will man eine komplette Liste ohne Lücken so startet man den Trace von der Knoten - Console aus und lässt den Trace auf die HD schreiben. Mit RUNBATCH MONI\_ON.TNB kann man den Trace starten und durch Aufruf einer MONI\_OF.TNB ihn wieder beenden. In der MONI\_ON.TNB leitet man den Monitor mit esc @E1 in ein File um und startet dann den Trace entsprechend. Mit MONI\_OF.TNB wird der Monitor wieder auf esc @E0 gestellt und der Trace mit TRACE 0 MN ausgeschaltet.

Beispiel: (TR)ace 9 (ALLE anzeigen, wirklich alles)

- 9 = reine Information
  - empfangene Nodes-Info
- 7 = wichtige Information
  - fastlearn <call> via <call>
- 5 = wichtiges Ereignis
  - unreachable source node <call>
  - own frame to <call> received fm <call> (loop)
  - frame fm <call> to <call> via <call> flushed (neigh loop)
- 3 = nicht kritischer Fehler
  - invalid node <call> > <call> from <call> (flushed)
  - invalid alias from <call> (flushed)
  - INP: frame error, len = <n>, left = <n>
  - destination <call> received from <call> (ignored)
- 1 = kritischer (fataler) Fehler
  - Frame rejected <call> -> <call> .....
  - send to disc'ed link <call> > <call> via <vialiste>
  - token lost <call>
- 0 = aus

Der Monitor erlaubt auch ONLINE ein Call oder Port zu monitoren, ohne es zuerst in ein File im Knoten zu schreiben.

MNLUSTIC H/F <port nr.> +/- <call..call>  
 M = wird ignoriert (kompatibilität zu älteren Versionen)  
 C = wird ignoriert und immer als eingeschaltet gekennzeichnet  
 (kompatibilität zu älteren Versionen und zum Monitor-Befehl  
 an der Konsole)

Monitoren von:

- N = Monitorfunktion ausgeschaltet.
- L = Die Länge des Info-Feldes (abzüglich L3/4-Header) wird bei I- und UI-Frames angezeigt
- U = Unprotokollierten Paketen,
- I = Info Paketen,
- S = Kontroll Paketen,
- T = Timestamp einschalten, also Datum und Uhrzeit werden mit angezeigt,
- H = HEADER, bei I- und UI-Frames nur den Header anzeigen, der Infoteil der Frames wird unterdrückt,
- F = FULL, I- und UI-Frames mit Infoinhalt anzeigen, es werden die kompletten Frames gemonitort,
- <port nr> = Nur der angegebene Port wird gemonitort,
- + <call..call> = Nur diese Call (bis zu 8) monitoren,
- <call..call> = Diese Call im Monitor unterdrücken.
- <call> kann mit SSID angegeben werden.
- Unter <call> sind auch Eingaben wie QST und LSTAT möglich.

Beispiel: (TR)ace 1 MUSICH 9 + DG0XX.

TNN meldet die Einstellung als Quittung zurück.



**TCPIP Kommandos**

**(ARP)** <DestIP> +/- <publ.> <Port> DG/VC <call[-\*> <[digi1> <digi2>]

Nimmt <destip> mit <publish> <port> als DC/VC als <call> in die Linkliste auf. Die Digipeater <digi1> und <digi2> können mit angegeben werden.

ARP-Einträge macht TNN automatisch. Sie müssen nur für Wege über via-Digipeater und NET/ROM-Verbindungen angegeben werden, da hier ARP nicht möglich ist.

ARP-Einträge werden für Gateways gemacht, bzw. für direkt erreichbare Stationen.

DestIP	Ist die IP-Nummer des (IP-) Nachbars auf diesem Port
+/-	Zum Ein- oder Austragen
Publ.	Publish soll dieser Port/Nachbar allen anderen auch bekannt gegeben werden? Ein P steht für „ARP beantworten“, wird es weggelassen, werden ARP-Anfragen an dieses Ziel ignoriert.
Port	Portname des Knotens über den die IP-Adresse erreichbar ist. Es kann hier auch wieder NETROM angegeben werden. Vorsicht! Schreibfehler im Portnamen führen evtl. zum Absturz (wir arbeiten noch daran).
DG/VC	Datagram/Virtual Circuit. Soll das ganze Verbindungslos aufgebaut werden (Datagram) oder soll eine Verbindung da sein (Virtual Circuit). Von Digi zu Digi sollte DATAGRAM eingestellt sein, dann gibt es weniger overhead. Dann findet allerdings auch keine Fehlerkorrektur auf dem Link statt, dies wird aber (zwar etwas langsamer) von den Endstationen (TCP/IP) gemacht. Auf DAMA-Einstieg ist UNBEDINGT VC einzustellen. Bei NETROMs hat dieser Parameter keine Bedeutung.
Call	AX-25-Call, unter dem der Rechner mit der IP-Nummer <destip> erreichbar ist.
Beispiel:	ARP 44.130.82.41 + P Sysop_9k6 VC DB7KG DB7KG-5

Was geht, und was nicht geht:

- Der IP-Router von TNN kann Frames beliebiger Länge durch Fragmentierung übertragen. Die Reassemblierung wird der Endstation überlassen.
- TNN empfängt aus Kompatibilitätsgründen zu FlexNet nur Pakete bis 256 Bytes Info. Dies wird sich zwischen zwei TNN in Zukunft ändern.
- ARP/REV ARP-Anfragen werden sofern erlaubt, beantwortet.

**(IPA)**dress 0.0.0.0

Setzt die IP-Adresse des Knotens.

**(IPR)**oute <destip>/<maskbits> +/- <interface> <gateip> <metric>

Nimmt <destip>/<maskbits> auf <intercace> <port> <gateip> in die IP-Routenliste auf.

DestIP	Ist die IP-Nummer des (IP-) Nachbars auf diesem Port
Maskbits	Hier wird entweder eine Zahl (Anzahl Bits) oder eine Art Pseudo IP-Nummer eingetragen. Anzahl der Bits heißt: Eine IP-Nummer hat 4 Bytes mit 8 Bit = 32 Bit. Trage ich hier 16 ein, so werden nur die ersten 2 Byte ausgewertet (entspricht für AFU: 44.130; 44=AFu 130=DL). Trage ich 24 ein, wird schon die Region beachtet, also 3 Bytes (44=AFU 130=DL 27=KS) bei einer (Beispiel) Nummer 44.130.27.80. So kann man hierarchisch Routen aussortieren. Bei der anderen Möglichkeit wird für das Byte, was frei bleibt eine 0 eingetragen.

Bsp.: 44.130.27.0 = Kassel.

Kann auch mit Broadcast zusammen so aussehen:

255.255.255.0 oder 44.130.27.255..kommt ganz darauf an

+/-	Zum Ein- oder Austragen
Interface	Hier wird die Schnittstelle benannt, wo die Pakete für <DestIP> herauskommen sollen. Dies kann der NAME eines Ports sein (also User_1k2, User_9k6). Es kann auch „NETROM“ angegeben werden. In diesem Fall wird das Packet an ein NETROM-Ziel (das muß natürlich auch ein Gateway [TNN] oder ein TCP/IP-Host [KA9Q] sein) übertragen.

GateIP	Wird ein Gateway benannt, so wird hier die IP-Nummer des Gateways eingetragen. Über ihn werden dann alle (IP-)Packets vermittelt.
Metric	Diese Angabe wird in einer späteren Version Bedeutung haben, z.Z. bitte ignorieren.
Beispiel:	IPROUTE 44.130.27.0/24 + User_9k6

**ESC/ALT Kommandos**

&lt;ALT&gt; X

Bricht das Programm ab und geht auf die DOS-Ebene zurück. Bei Linux geht das nur mit <ESC> QUIT.

Die <ESC>-Befehle (<ESC> steht hier für die entsprechende Taste auf der Konsolentastatur) steuern die Konsolenfunktionen. Einige dieser Befehle können auch per Fernbedienung eingegeben werden (s. Befehl ESC).

&lt;ESC&gt; B

Anzeigen Rounds/s

&lt;ESC&gt; C

Connect von der Tastatur als HOST in den Knoten. Im Terminalmodus ist man automatisch als Sysop privilegiert.

&lt;ESC&gt; D

Disconnect vom Knoten

&lt;ESC&gt; @E1

Öffnet ein neues Protokollfile YYMMDDHH.PRO. Wenn das File bereits existiert, wird es nicht gelöscht, sondern die Ausgaben werden ans Ende angehängt. Die Monitorausgaben des Batch-Kanals werden in dieses Protokollfile umgeleitet. ACHTUNG! Die Ausgaben auf dem Konsolenbildschirm werden NICHT mehr in das Protokollfile geschrieben. Das geht nur noch von einem TNB-File aus (in dem der passende TRACE-Befehl eingetragen ist).

&lt;ESC&gt; @E0

Schaltet die Ausgabe ins Protokollfile ab und schließt das File.

&lt;ESC&gt; I &lt;call&gt;

Einstellen des HOST Calls für den aktiven Hostkanal.

&lt;ESC&gt; I &lt;call&gt; &lt;alias&gt;

Setzen des globalen Host-Calls, wenn <call> nicht mit dem TNN-Rufzeichen übereinstimmt. Gleichzeitig mit dem Setzen des Calls wird zusätzlich ein Link-Eintrag für <call> mit <alias> vorgenommen auf dem (virtuellen) Port 16 (bei der 16 Kanal-Version). <call> und <alias> können dann vom User connected werden, entweder mit dem Befehl „C“ ohne Parameter oder mit dem entsprechenden Call als Parameter. Soll der Linkeintrag wieder gelöscht werden, wird das TNN-Rufzeichen eingegeben, aber auch hier wird ein <alias> benötigt (der aber ignoriert wird).

&lt;ESC&gt; K

Anzeigen der Einstellung für die Datums-/Zeitanzeige bei Statusmeldungen.

&lt;ESC&gt; K0

Statusmeldungen werden ohne Datum und Uhrzeit dargestellt.

&lt;ESC&gt; K2

Statusmeldungen werden mit Datum und Uhrzeit dargestellt.

&lt;ESC&gt; Logout

Verlassen der Konsole und verschließen.

&lt;ESC&gt; L

Zeigt den Status der (30) Host-Kanäle an.

&lt;ESC&gt; M

Zeigt die derzeitige Einstellung an. (Bedeutung der Parameter siehe unten)

<b>ACHTUNG !</b>	Bei den Monitorbefehlen ist, bei stark belasteten Knotenrechnern, Vorsicht geboten. Da der Rechner dann um einiges langsamer wird, bekommt man das Monitoren nur schwer wieder ausgeschaltet, da die Tastatur kaum noch abgefragt wird.
<ESC> MN	Monitor aus. (Je mehr der Monitor anzuzeigen hat, desto langsamer wird der Knoten. Also nach Gebrauch immer den Monitor wieder auf „AUS“ schalten !
<ESC> MLUSTIC H/F <port nr.> +/- <call..call>	(Siehe oben ESC MLUSTIC H/F <port nr.> +/- <call..call>)
<ESC> QUIT	TNN beenden.
<ESC> R0	Es erfolgt kein Hinweis auf Token-Recovery.
<ESC> R1	Bei jedem Token-Recovery wird auf dem Bildschirm der Text *** Token-Recovery (1) Wed Jan 19 13:45:10 1994 *** angezeigt. Die Ziffer gibt an, wieviele Recovery nacheinander aufgetreten sind, sowie Datum und Uhrzeit. Dieses ist die default Einstellung.
<ESC> S	Zeigt den Aktuellen Host-Kanal an.
<ESC> Sn	Schaltet zum Kanal n um. (Gültige Kanäle sind 1 - 30)
<ESC> T	Zeigt die momentan auf dem Tokenring eingestellte Baudrate. Dieser Befehl gilt nicht für die Linux-Version.
<ESC> T 9600 / 19200 / 38400	Stellt die entsprechende Baudrate auf dem Tokenring ein.
<ESC> T 57600 / 115200	Stellt die entsprechende Baudrate auf dem Tokenring ein, für diese Baudraten ist jedoch ein FIFO-Baustein (16550) erforderlich.
<ESC> T ?	Gibt eine Hilfe aus.
<ESC> V	Anzeigen der Hostmode-Version.
<ESC> Y0	Kein Connect zum HOST möglich (default).
<ESC> Yn	Verbindungen zum HOST sind auf n Hostkanälen freigegeben (n = 1 bis 30). Der Connect zum HOST ist mit C möglich <b>OHNE</b> den Zusatz eines <call>, sowie mit C <CALL>, wobei für <CALL> das globale Host-Call gilt.
<ESC> @B	Anzeige der für den Hostmode verfügbaren freien Buffer (1/4 der Buffer am Knoten).
<ESC> @Y	Anzeigen der maximal verwendeten Host-Kanäle.

&lt;ESC&gt; @Y&lt;n&gt;

Einstellen der maximal zu verwendenden Host-Kanäle (für den Terminalmodus oder ein Hostmodeprogramm, das nicht mit den 30 Host-Kanälen arbeiten kann / soll. Hier ist zu beachten, daß der aktive Kanal in dem erlaubten neu zu setzenden Bereich liegt, also wenn der aktive Kanal 20 ist, kann man nur auf 20 .. 30 Host-Kanäle einstellen, nicht aber auf 19. Ist einer der in Zukunft nicht mehr zu verwendenden Kanäle noch connected, wird er automatisch disconnected. Anschließend werden mit dem Befehl <ESC> L nur noch die verwendbaren Kanäle angezeigt und man kann auch nicht auf einen Kanal mit höherer Nummer umschalten (mit <ESC> S).

**(HOST)mode-Befehle**

Im Hostmode stehen (fast) die gleichen Befehle zur Verfügung, wie im Terminalmodus. Es wird allerdings das von TF her bekannte Hostmode-Protokoll verwendet. Folgende Hostmode werden von TNN 1.76 unterstützt:

C

Connect zum Knoten.

D

Disconnect vom Knoten.

E

wird ignoriert (wegen Kompatibilität zu bestehender Software).

G

empfangene Info-, Monitor- und Status-Daten abholen.

I

anzeigen / setzen des Calls für den aktiven Kanal.

K

Statusmeldungen mit / ohne Uhrzeit.

L

Anzeigen der für den aktiven Kanal vorliegenden Infos.

M

Monitor ein / aus (Parameter s. TRACE).

V

Hostmodeversion anzeigen.

Y0

Kein Connect zum HOST möglich (default).

Yn

Verbindungen zum HOST sind auf n Hostkanälen freigegeben (n = 1 bis 30). Der Connect zum HOST ist mit C möglich **OHNE** den Zusatz eines <call>, sowie mit C <CALL>, wobei für <CALL> das globale Host-Call gilt.

@B

Anzeige der fuer den Hostmode verfügbaren freien Buffer (1/4 der uffer am Knoten).

@S

Anzeigen Linkstatus (für Kompatibilität mit bestehender Software).

@Y

Anzeigen der maximal verwendeten Host-Kanäle.

@Y<n>

Einstellen der maximal zu verwendenden Host-Kanäle (für den Terminalmodus oder ein Hostmodeprogramm, das nicht mit den 30 Host-Kanälen arbeiten kann / soll. Hier ist zu beachten, daß der aktive Kanal in dem erlaubten neu zu setzenden Bereich liegt, also wenn der aktive Kanal 20 ist, kann man nur auf 20 .. 30 Host-Kanäle einstellen, nicht aber auf 19. Ist einer der in Zukunft nicht mehr zu verwendenden Kanäle noch connected, wird er automatisch disconnected. Anschließend werden mit dem Befehl <ESC> L nur noch die verwendbaren Kanäle angezeigt und man kann auch nicht auf einen Kanal mit höherer Nummer umschalten (mit <ESC> S).

**(#####.TNB) FILES****....ein Leckerbissen**

Ein ganz alltägliches Problem: immer nachts um 03:00 passiert angeblich Fürchterliches am Knoten, wenn der OM AA0BB erscheint. Man müßte mal am Knoten mitmonitoren. Oder man möchte jede Nacht um 23:00 die Statistik auf Disk schreiben. Oder von 16:00 bis 24:00 sollen andere Parameter auf dem Einstieg genommen werden. Alles Beispiele für den Einsatz von TNN-Batch Files.

Die Sache ist ganz einfach. Es gibt zwei Sorten von Batchfiles, die sich durch den Namen unterscheiden: YYMMDDHH.TNB und YYMMWXHH.TNB. Dabei ist YY das Jahr, MM der Monat, DD der Tag, HH die Stunde und X der Wochentag, an dem das File gestartet werden soll. Und damit nicht für jede Aktion ein extra File gemacht werden muß, darf „#“ als Platzhalter für eine Ziffer genommen werden.

Verzeichnis:

Die .TNB Dateien gehören in das Verzeichnis, welches in der TNN176.PAS unter Workpath angegeben wurde.

Beispiele:

TNN176.TNB	Wird NUR beim Neustart des Programmes ausgeführt.
#####.TNB	Wird beim Neustart sowie zur vollen Stunde ausgeführt.
#####23.TNB	Startet jeden Tag um 23:00 Uhr.
####0100.TNB	Startet immer am 1. jedes Monats um 00:00 Uhr.
99040100.TNB	Startet am 01-APR-99 um 00:00 Uhr.
####W116.TNB	Startet immer Montags um 16:00 Uhr (Sonntag ist Tag 0).
99##W011.TNB	Startet an jedem Sonntag im Jahre 1999 um 11:00 Uhr.

Der Aufbau der Files ist simpel: es steht alles so drin, wie man es auch an der Console eintippen würde. Mit einer einzigen Ausnahme: wenn ein Batch abläuft, ist automatisch ein Login passiert, und nach Ablauf des Files ist der alte Login Status wieder vorhanden.

Damit nun nicht nur der Bildschirm am Knoten gefüllt wird, sondern auch was für die Nachwelt bleibt, wird mit dem Befehl <ESC> @E1 ein Protokoll-File geöffnet bzw. mit <ESC> @E0 wieder geschlossen (siehe oben).

Um eine monatliche Statistik zu erzeugen, ist folgendes File nötig:

```
####0100.TNB
=====
;                Dieses File startet am 1. des Monats um 0:00 Uhr.
esc @e1          ; Umlenken der Ausgabe in ein File.
Stat             ; Auslesen der Statistik.
MH 5000 +        ; Auslesen der MH Liste.
Esc @e0          ; Umlenken ausschalten.
Clear            ; Statistik löschen.
MH -             ; USER-Statistik auf „0“ Stellen.
```

Ab dem „;“ sind Kommentare erlaubt.

Genauso sind auch DOS Kommandos möglich. Bei der Linux-Version sind SHELL Kommandos zwar zulässig, aber sie werden meist nicht ausgeführt. Der Grund hierfür: SHELL Kommandos werden im Hintergrund ausgeführt als eigener Prozeß (der normale Knotenbetrieb läuft also weiter). Wird während eines laufenden SHELL Kommandos ein neuer Befehl eingegeben (das ist der nächste Befehl im TNB-File), wird das laufende Kommando abgebrochen und der neue Befehl wird ignoriert.

**(RUNBATCH) <filename.tnb>**

Hiermit lassen sich aus einer TNB weitere TNB aufrufen. Damit ist es möglich die TNN176.TNB z. B. in eine PARAM.TNB und eine IPNUMMER.TNB zu unterteilen.

**EXTERNE PROGRAMME/BEFEHLE für den SYSOP:****(O)utput** (externer Befehl)

OUTPUT.EXE ist nun ein eigenständiges Programm geworden, das der Sysop im Bedarfsfall aufrufen kann. Der Aufruf erfolgt wie bei einem internen Befehl. OUTPUT.EXE ist, wenn es sich im PATH SYSEXEC befindet, NUR vom SYSOP nach der Privilegierung aufrufbar. Bei Eingabe von Output ohne Argument wird der momentane Zustand der Portbits angezeigt.

KS:DB0EAM> IO-Port-Status								
0	1	2	3	4	5	6	7	
1	1	1	1	1	1	1	1	

Der Befehl Output ist für Fernsteuerungen vorgesehen. Die 8 Datenleitungen des Druckerports (LPT1) des PC können bitweise ein- und ausgeschaltet werden.

Der Ausgangszustand nach einem Programmstart ist „**alles eingeschaltet**“. Die Basisadresse des Ports wird über das Rechner-BIOS abgefragt. Soll also ein anderer Port als LPT1 verwendet werden, so muß nur bei 40:08 die passende Adresse stehen.

**(O)utput** <port\_bit> <ein\_aus>

<port\_bit> : zu schaltendes Datenbit (0..7).

<ein\_aus> : neuer Pegel (0 oder 1).

**(CWER)** (externer Befehl)

Ist ein kleines Programm, welches die Visitenkarten der User, die sie sich selbst mit /P im Conversmode erzeugt haben, anzeigt.

**(CWER)** a

Erzeugt eine Liste mit allen Einträgen.

**(CWER)** l <anzahl>

Zeigt die l=letzten <anzahl> von Einträgen an (die neuen werden immer hinten angehängt).

**(CWER)** <call>

Kann auch in Verbindung den Platzhaltern „\*“ und „?“ aufgerufen werden. Dabei steht „\*“ für beliebig viele Zeichen und „?“ für genau 1 Zeichen.

**(CPERS)** (externer Befehl)

Ist nun nur noch zur Pflege der „Visitenkarten“ der Datei CONVERS.PRS. Zum einen gibt es mitunter mal Einträge, die „zufällig“ entstanden sind, und die der Sysop gerne entfernen möchte. Oder sie auch nur durch einen anderen ersetzen. Der einfache Aufruf des Programmes entfernt „Leereinträge“ ohne Text sowie „unsinnige“ Texte, die zum Beispiel nur aus einem Zeichen bestehen. Bei jedem Aufruf des Programmes wird eine Datei CONVERS.ALT angelegt, um zumindest die letzte Änderung der Datei noch einmal rückgängig machen zu können. Da es durch den Aufruf dieses Programmes zu Speicherplatzproblemen (CONVERS.PRS zu groß) kommen kann, überprüft das Programm zuerst, ob die Datei noch in den verbleibenden Speicher geladen werden kann.

**(CPERS)** /sort

Sortiert die Einträge alphabetisch nach dem Call.

**(CPERS)** + <call> <text>

Ersetzt bzw. fügt das Call mit dem <text> an die Datei CONVERS.PRS an. Es wird hierbei **KEIN** Callcheck durchgeführt !

**(CPERS)** - <call>

Entfernt das Call und den Eintrag aus der Datei CONVERS.PRS. Auch hier **KEIN** Callcheck, damit auch die unsinnigsten Einträge bearbeitet werden können.

**(SETCALL)** (externer Befehl)

Die Felder können mit (SETCALL) ausgefüllt oder geändert werden. SETCALL soll dem SYSOP ermöglichen auf die Datensätze zugreifen zu können.

Zu den Programmen SHOWCALL.EXE, SAVECALL.EXE und SETCALL.EXE ist in dem Dir SAVECALL jeweils eine Datei mit der Endung .SXX. Werden diese Sprachdateien in .SPK umbenannt, so werden sie statt der internen Texte benutzt, was jedoch einen zusätzlichen HD-Zugriff bedeutet. Sollte jemand diese Dateien übersetzen, so würde ich mich über einen Rücklauf freuen.

**(SETCALL)** <call> <feldkennung> <text>

```

-> SaveCall Version [010696] de DG3AAH <-
SETCALL CALL /N ..... 30 Zeichen für den Name
SETCALL CALL /Q ..... 30 Zeichen des Wohnortes
SETCALL CALL /L ..... 6 Zeichen des World-Locators
SETCALL CALL /D ..... 7 Zeichen für Eingabe des DOK
SETCALL CALL /V ..... 9 Zeichen QRV auf ... Digi
SETCALL CALL /M ..... 20 Zeichen Mybbs der Mailbox
SETCALL CALL /T ..... 40 Zeichen für freien Text
    ( ein * als Text löscht diesen Eintrag )

SETCALL CALL /*      : löscht den Inhalt aller Eintrags für Call.
SETCALL CALL /B      : blockiert CALL für ShowCall.
SETCALL CALL /F      : gibt Call für ShowCall frei.
SETCALL CALL         : gibt Infos über Call aus, auch wenn blockiert.
    ( ein B als Call gibt alle blockierten Calls aus )

```

Auf der Diskette ist jeweils ein Grundbestand von Calls vorhanden (der von DB0EAM). Updates davon gibt es nur gegen Zusendung des eigenen Datensatzes und wenn sich jemand findet der eine SORTIERROUTINE sowie die Erzeugung der entsprechenden CALL.IDX dafür schreibt.

**(SYST)ext Text** (externer Befehl)

Systext bietet nun die Möglichkeit, in die Datei SYSOP.PRO eine Zeile Text einzufügen. Damit können zum Beispiel Parameteränderungen für die „anderen“ Sysop dokumentiert werden. Damit ist nun auch eine gewisse Historie möglich. Die Datei SYSOP.PRO wird übersichtlicher, wenn die Einträge anders ausgegeben werden. Auch Neustarts (z.Z. mit b-log.exe dokumentiert) sollen noch in der SYSOP.PRO aufgenommen werden.

**(SH)owSYS** (externer Befehl)

SHOW\_SYS zeigt die Einträge in der Datei SYSOP.PRO in einer übersichtlichen Form an.

**(SH)owSYS /l**

SHOW\_SYS zeigt die Einträge in der Datei SYSOP.PRO in einer übersichtlichen Form an und löscht dabei alle Privilegierungs-Einträge.

**(MSY)** (externer Befehl)

MSY.EXE in SYSEXEXE ist ein Hilfsprogramm für den Sysop, das zu MSG.EXE in USEREXE gehört.

**(MSY) C**

Lifetime aller Nachrichten wird um 1 verringert. Dieses kann per Hand geschehen oder mit einem solchen TNB-File.

```

#####01.TNB
=====
;           ;Diese File startet täglich um 1:00 Uhr.
MSY C      ;Lifetime der .MSG-File um einen Tag herabsetzen
;           ;und bei Lifetime = 0 entfernen.

```

**(MSY) D**

Listet die Header aller Nachrichten auf.

**(MSY) D -#x o. +#x**

Nachrichten, deren Lifetime (-) kleiner o. (+) größer x-Tage ist.

**(MSY) D -\*x o. +\*x**

Nachrichten, die (-) jünger o. (+) älter als x-Tage sind.

**(MSY) D FROM**

Zeigt die Nachrichtenliste sortiert nach Absender Calls an.

Die Sortierung kann auch nach :  
BYTE, DATE, TIME, LT, RETURN vorgenommen werden.



(MSY) E ALL <call>

Löscht alle Nachrichten von oder an <call>.

(MSY) E 01.01.95/00:00:01

Löscht alle Nachrichten dieses Datums/dieser Uhrzeit.

(MSY) G <gruppe> +/- <call>

Fügt/löscht <call> aus <gruppe>. Zugelassen in einer Gruppe sind maximal 50 Call.

(MSY) L

Zeigt alle Nachrichtenfiles.

(MSY) S <call> #x <text>

Schreibt <text> an <call> mit Lifetime #<x>. Hier ist eine Lifetime von maximal 99 Tagen möglich.

(MSY) V

Ausgabe von der Versionsnummer und dem Datum.

(SYSH)elp (externer Befehl)

Ist ein externer Befehl für eine neue Hilfefunktion. Funktion wie bei (H)elp nur wird auf die OHS.TXT Online-Hilfe-Sysop zugegriffen.

(TOP) (externer Befehl)

Ist ein externer Befehl, der die MHEARD.TAB in besser lesbarer Form anzeigt. In der CONFIG.TOP müssen allerdings die Port - Namen eingestellt werden. Da nun auch die Level 3 Verbindungen geführt werden, sollte man alle Port - Namen dort eintragen.

```
# Configfile für TOP.EXE.
# Alle Angaben die nicht gelesen werden sollen müssen eine # am Zeilenanfang haben.
#
# Einstieg
P0 = 70cm_1200
# Einstieg
P1 = 70cm_9600
# Einstieg
P2 = 23cm_9600
# Link DB0GOE
P3 = Goettingen
# ... bis
p16 = ???
# Ende des Files
```

(STARTCNT) (externer Befehl)

eine sehr hilfreiches Programm.

Wer häufiger neue Testversionen an seinem Digi einsetzt oder mit den TNN176.TNB Konfigurationsdateien spielt, der sollte unbedingt die Utility "STARTCNT.EXE" verwenden! Dieses Programm wird zusammen mit den TNN-Tools verteilt und stammt von Andreas, DB7KG. STARTCNT ist ein kleines EXE-Programm, welches einen Zähler bei jedem Aufruf herunterzählt. Solange der Zähler nicht null ist, wird immer ein ERRORLEVEL von 1 zurückgegeben. Ist der Zähler abgelaufen, dann ist der ERRORLEVEL 0. Damit lassen sich dann mittels einer Batch-Datei sehr bequem zwei verschiedene Versionen starten. Damit STARTCNT auch bei Stromausfall funktioniert, wird der Zähler natürlich in einer Datei gespeichert.

Wenn man sich als Sysop in den Knoten eingeloggt hat, dann kann man z.B. mit "DOS STARTCNT 5" diesen Zähler auf 5 setzen. Nun darf der Knoten 5mal abstürzen / resetten oder was auch immer, bis wieder die alte (stabile) Software gestartet wird.

STARTCNT <anzahl>

stellt die jeweilige Anzahl an Starts ein (Durchläufe der START.BAT). Die Anzahl wird um 1 verringert, wenn die STARTCNT.EXE ohne <anzahl> aufgerufen wird.

Die folgenden Zeilen müssen dann jedoch in die START.BAT aufgenommen werden.

```
STARTCNT
IF ERRORLEVEL==1 GOTO OKGNU32
ECHO Starte die TNN_ALT.EXE

REM !!!!!!!!!!!!!!! ALTE SOFT HIER STARTEN (TNN_ALT.EXE) !!!!!!!!!!!!!!!
COPY TNN176.DOS TNN176.TNB
TNN_ALT.exe
REM !!!!!!!!!!!!!!! ALTE SOFT HIER STARTEN (TNN_ALT.EXE) !!!!!!!!!!!!!!!

GOTO ENDE
:OKGNU32

ECHO Starte die TNN_NEU.EXE

REM !!!!!!!!!!!!!!! HIER WIRD DIE NEUE SOFTWARE GESTARTET !!!!!!!!!!!!!!!
COPY TNN176.DPI TNN176.TNB
TNN_NEU.exe
REM !!!!!!!!!!!!!!! HIER WIRD DIE NEUE SOFTWARE GESTARTET !!!!!!!!!!!!!!!
```

**HILFSPROGRAMME für den SYSOP zu Hause:****(MAKEDAT.EXE)**

Wer Sat-Daten auf dem Digipeater anbietet, muß diese auch pflegen! (Andernfalls solltet Ihr lieber die SAT.EXE in USEREXE löschen.)

DB1HZ hat nun statt der bisher 22 Satelliten 40 Berechnungen ermöglicht in der neuen SAT.EXE. Außerdem stehen die Daten nun nicht mehr in Einzelfiles, sondern es ist nun nur noch das File SAT.DAT im Verzeichnis TNN zu pflegen. Das Verzeichnis DATEN kann somit entfallen. Ferner wird nun ein evtl. vorhandener CO-Prozessor unterstützend bei der Berechnung hinzugezogen, was die Berechnungszeiten positiv beeinflusst. Die bisher vorhandene Umgebungsvariable SET SATQTH=<Locator> oder <Koordinaten> ist entfallen. Es wird nun die bereits vorhandene Variable QTH mitbenutzt.

Das Programm MAKEDAT dient der Umwandlung der via Packet-Radio verbreiteten Keplerdaten im AMSAT- oder 2LINE-NASA-Format in die von SAT Master erwarteten Datensätze. Nähere Informationen zu MAKEDAT bekommst Du, indem Du das Programm ohne Parameter aufrufst.

Zum Ablauf:

Man öffnet auf seinem Rechner zu Hause ein SAVEFILE z.B. mit dem Namen KEPLER.TXT. Dann kann man mit <read kepler> in der Box alles, was an neuen Daten Texten und Programmen dort vorhanden ist, einlesen. Nach dem Schließen des SAVEFILES wird nun MAKEDAT /I KEPLER.TXT aufgerufen. MAKEDAT aktualisiert nun nur die Einträge der SAT, die in der Datei SAT.FLT eingetragen sind, in der Datei SAT.DAT. Anschließend die Datei SAT.DAT <load sat.dat> wie ein Programm in den Knoten laden .... fertig ist es!

**(TNNSET.EXE)**

Stammt von IK7NXQ und soll das Erstellen von EPROMS für der Tokenring erleichtern. Einfach mal aufrufen .. es erklärt sich selbst.

**Befehle für alle User****(BE)ACON**

Zeigt die eingestellten Bakenparameter und Bakentexte an.

**(C)ONNECT**

Connect zum HOST-Interface wenn dieses mit <ESC> **Y 1** freigegeben ist. Ist Y auf 0 gesetzt, so wird am Terminal ein CONNECT REQUEST fm <call-ssid> <datum uhrzeit> angezeigt.

**(C)ONNECT <call>**

Mit dem Connect-Befehl wird eine Verbindung zu einem anderen Knoten oder einem anderen Benutzer aufgebaut. Die Eingabe

**CONNECT DB0FD**

oder auch abgekürzt und klein geschrieben

**c db0fd**

bewirkt, daß erst in der NODES-Liste (Nodes und Flexnet-Destinations) nach dem Call DB0FD abgesucht wird. Handelt es sich um ein Call, das in der NODES-Liste eingetragen ist, so wird die Meldung:

**Interlink setup (via call) ...**

ausgegeben und in Klammern wird angezeigt zu welchem direkten Nachbarn der Connect gesendet wird.

Wird dieses Call lokal geführt, so wird ein:

**Link setup (Portname)...**

Ist kein Eintrag vorhanden, so wird die MH-Liste durchsucht. Ist dort ein Eintrag vorhanden, so wird der Connect auf dem entsprechenden Port ausgesendet. Ist auch dort kein Eintrag vorhanden, so wird eine Fehlermeldung

ausgegeben. An den Meldungen ist bereits erkennbar, und somit auch durch Router auswertbar, ob eine Verbindung zu einem TheNet - bzw. TheNetNode-Knoten überhaupt aufgebaut werden kann.

Ist der zu verwendende Port nicht eingeschaltet, so wird angezeigt:

**Port not in use**

Wurde eine Verbindung zu einem TheNet - oder TheNetNode-Knoten aufgebaut, wird die Meldung mit dem ALIAS ausgegeben, sofern vorhanden:

**BS:DB0FC> Connected to H:DB0FD** ansonsten nur :  
**BS:DB0FC> Connected to DB0FD**

Ein Benutzer kann auch über Digipeater connected werden. Die Eingabe ist dann:

CONNECT DB3AN via DB0FD

Das Wort via kann entfallen oder beliebig abgekürzt werden. Es sind auch bis zu 8 Digipeater möglich. Die Rufzeichen sind dann durch Leerzeichen zu trennen. Diese Möglichkeit funktioniert aber nur, wenn bei dem entsprechenden Digipeater auch L2-Digipeating eingeschaltet ist.

Mögliche Fehlermeldungen sind:

Failure with ... : Der gesuchte Partner hat sich nicht gemeldet.

Busy from ... : Der Partner hat einen Verbindungsaufbau abgelehnt.

Node not available : Das Ziel ist zwar bekannt aber es gibt zur Zeit keine Verbindung dahin.

Die folgenden Meldungen zeigen an, daß der Knoten, an den der Connect-Befehl geschickt wurde, überlastet ist und daher die Verbindung nicht aufgebaut werden konnte:

Node busy  
Link table full  
Circuit table full

Ein Connect-Befehl kann jederzeit durch eine beliebige Eingabe abgebrochen werden, eine Leerzeile reicht aus.

Im Connect-Befehl kann auch der Port-Name mit angegeben werden. Gültige Port-Namen erhält man mit dem (PO)rt-Befehl. Auf dem Multibaud Zugang von DB0EAM sind u.a. die Port-Namen „70cm\_1200“, „70cm\_9600“ und „23cm\_9600“ eingetragen. Wird im Connect-Befehl der Port-Name mit angegeben, so wird der Connect auf diesem Port ausgesendet ohne Beachtung eines Eintrages in der MH-Liste. Format: C DB6XYZ 23cm\_9600.

Wird ein Connect ohne Argument eingegeben, so wird versucht, eine Verbindung zum HOST-Terminal herzustellen. Das ist jedoch nur möglich, wenn der Sysop mit dem Befehl <ESC> y1 Verbindungen zum HOST-Terminal freigegeben hat und davon eine nicht belegt ist, ansonsten gibt es ein :

Busy from <call>.

LOOP DETECTED ist eine Warnung, daß der gewählte Verbindungsaufbau eine Schleife (Loop) bildet. Es wird also eine Interlink-Verbindung in beiden Richtungen über den selben Nachbarknoten und somit unnötig belegt. Im Gegensatz zu anderer Knotensoftware erfolgt jedoch nur eine WARNUNG, die Verbindung kann trotzdem aufgebaut werden.

WARNING: loop detected via neighbour: DB0.. (Help Loop)

Letzteres ist aber nicht immer sinnvoll und man behindert sich selbst.. Bitte mit 2 mal Quit zum vorletzten Knoten zurück gehen und dann die Verbindung direkt aufbauen... Es kann aber auch vorkommen, daß der Router von TNN einen neuen Weg zum Zielknoten gefunden hat, der zu der Schleife führt — dann kann man die Warnung auch einfach ignorieren.

#### **(CONV)ers**

Schaltet um in den Convers-Modus.

#### **Der Convers-Modus bei TheNetNode**

Seit Conversd v2.14 besteht die Möglichkeit, Convers-Knoten untereinander zu vernetzen, d.h. ein Convers-User muß sich nicht über einen langen Digipeaterweg bis zu dem Convers-Knoten connecten, auf dem sich seine gewünschten Gesprächspartner befinden, sondern es genügt, wenn ein Connect zum nächsten Convers-Knoten aufgebaut wird, der das Conversd-Protokoll unterstützt. Leider neigte diese Version zu Schleifen beim Verbindungsaufbau.

Seit TheNetNode V1.50 (PC) ist ein wesentlich verbesserter Convers-Modus implementiert worden. Es handelt sich dabei um Conversd PingPong-Release 3.12, das von DK5SG entwickelt, von DC6IQ weiter geführt und von DL1XAO in TheNetNode eingebaut wurde.

Mit der v3.12 (pp-conversd) funktioniert die Vernetzung nun ohne Probleme. Außerdem wird eine Menge zusätzlicher Komfort geboten, wie z.B. Personaltexte, Kanalthemen, Kanaloptionen und Umlautwandler.

Auch ältere Conversd können angebunden werden, nur verstehen diese nicht alle Kommandos bzw. nicht vollständig, sie wirken praktisch als Filter für die neuen Funktionen.

Zwischen den Convers-Hosts werden alle Texte, welche die verschiedenen User schreiben, nicht mehr getrennt für jeden Benutzer einzeln, sondern nur noch einmal übertragen. Dies entlastet die Linkstrecken spürbar, da z.B. 10 Benutzern ein und dasselbe Packet über die Interlinks nicht 10mal übertragen werden muß, sondern nur noch 1mal!

Weiterhin ist der Convers-Einzugsbereich natürlich wesentlich größer geworden, und man kann davon ausgehen, daß man nun häufiger einen Gesprächspartner im Convers findet.

Die Convers-Befehle stehen unter **-(CONV)ERS - intern -**.

#### **(CONV)ers <n>**

Schaltet um in den Convers-Modus auf Kanal <n>.

**(CONV)ers C(stat)**

Zeigt die bestehenden Verbindungen, Laufzeiten, Datenmengen usw. an.

Host	State	Quality	Revision	Since	NextTry	Tries	Queue	RX	TX
db0goe	Connected	1s/1s	pp-3.06t	17:55			0	136K	267K
(DB0GOE on port 6)									
db0ii	Connected	10s/7s	pp-3.06t	7:07			0	1K	48K
(DB0II on port 9 via DB0BID)									
db0kh	Connected	1s/1s	pp-3.06t	6:13			0	17K	65K
(DB0KH on port 11)									
db0gso	Disc./locked	---		9:22	10:22		0		
(DB0GSO on port 9 via DB0BID)									
1 loops detected.									
dg9fu	Disconnected	---		18:47					
(trusted host)									
DB0KV	(dw-0.84k)	2m	DB0NOE	(dw-0.82b)	5m	db0acc	(pp-3.12f)	25s	
db0ais	(pp-3.12f)	3m	db0ber	(pp-3.06t)	3m	db0bhv	(pp-2.93t)	1m	
db0bid	(pp-3.06t)	9s	db0bro	(pp-3.06t)	12s	db0cl	(pp-2.93t)	2m	

Wird nun ein Convers-LOOP bemerkt, so wird der ermittelte Link für eine Stunde aus dem Verkehr gezogen. Trotzdem sollten LOOP vermieden werden. Die Anzeige der Einträge wie (DB0GOE on port 6) sind erst nach der SYSOP-Privilegierung sichtbar.

**(CONV)ers O(nline) [q ; l ; a]**

Zeigt die Benutzer das Convers an. Zusätzliche Optionen wie [q ; l ; a], sind möglich.

**(CONV)ers - intern -**

Im Convers-Modus stehen folgende Kommandos zur Verfügung (die Kommandos können durch die Verwendung der Großbuchstaben abgekürzt werden):

/Away [Text]	Markiert Dich als abwesend.
/All [Text]	Text an alle User Deines Kanals.
/Beep	Beep-Modus an/aus.
/Channel [n]	Verbindet dich zusätzlich mit dem Kanal n.
/CHARacter	Setzt verschiedene Zeichenwandlungen.
/Destinations	Listet erreichbare ping-pong Hosts.
/Help [Kommando]	Gibt Hilfe-Informationen.
/EXclusiv User Text	Sendet Text an alle auf Deinem Kanal außer User.
/Filter [Calls]	Setzt Calls, deren Texte gefiltert werden sollen.
/Invite User	Lädt User auf Deinen Kanal ein.
/Links [args]	Listet oder setzt (Sysop) conversd-Partner.
/LISt	Listet alle Kanäle und ihre Themen.
/LEave [Kanal]	Verläßt Kanal oder derzeitigen Kanal.
/Msg User Text	Sendet Text an User oder.
/Msg #Kanal Text	Sendet Text an den angegebenen Kanal.
/ME Text	Sendet einen Aktionstext.
/MOde [Kanal] Optionen	Setzt Kanalooptionen.
/NOtify [Calls]	Setzt Calls, deren Erscheinen gemeldet werden soll.
/Personal [Text]	Setzt persönliche Beschreibung.
/PRompt abcd	Prompt setzen (a=Query; b=Normal; c=Ctrl-g; d=Ctrl-h).
/Quit	Convers verlassen.
/QUery [User]	Startet/beendet private Konversation.
/Topic [#Kanal] [Text]	Setzt Thema des Kanals. Thema=@ entfernt Thema.
/UPtime	Wie lange läuft dieses conversd überhaupt schon ?
/Verbose	Laber-Modus ein/aus.
/VERSion	Zeigt Info zu dieser Version.
/Who [*; A; L; Q]	Zeigt User und Ihre Kanäle (*=eigner; A=abwesend; L=lang; Q=kurz).
/Width [Wert]	Setzt/zeigt Zeilenbreite.

**Die Erklärungen im einzelnen:****/ALL**

Wenn Du im /query Modus bist, wird Text mit vorangestelltem /all behandelt, als würdest Du ohne /query arbeiten.

**/AWAY <text>**

/away setzt den Abwesendheitstext, den die anderen lesen können. Beim Aufruf ohne Argument wird der Text gelöscht und man gilt wieder als anwesend.

**/BEEP**

Hiermit wird das Klingelzeichen (CTRL-G), welches vor jeder Mitteilung gesendet werden kann, ein- oder ausgeschaltet. Dieses Kommando ist eigentlich eine Untermenge des /prompt Befehls, siehe dort.

**/CHAN**

Verbindet Dich zusätzlich mit dem gewünschten Kanal. Im Gegensatz zu älteren Conversd-Implementationen verbleibt man auch noch im vorherigen Kanal, denn es wird eine Mehrfach-Kanal-Verbindung unterstützt. Um einen Kanal zu verlassen, mußt Du „/leave“ verwenden. Ohne Angabe eines Kanals wird die Info ausgegeben, auf welchen Kanälen Du Dich befindest.

**/CHAR**

Mit diesem Befehl kannst Du dem Convers mitteilen, welche Zeichensatzwandlung Du haben möchtest. Die Syntax ist: /char [In-Typ [Out-Typ]] wenn Du z.B. mit einem Atari ST arbeitest, könntest Du „/char pc atari“ eingeben. Wenn Du einen PC benutzt und Umlaute im TeX-Stil schreiben möchtest, gebe „/char tex pc“ ein. „/char ?“ listet die möglichen Einstellungen.

Die Einstellung wird mit „/pers“ gespeichert (siehe dort).

Der Dank für diese nette Funktion geht an Tommy, <dl9sau@thynet.sub.org> (Internet mail) <dl9sau@db0sao.ampr.org> (AmPR-Net mail). Vorschläge sollten an ihn weitergeleitet werden.

Mögliche Einstellungen mit /char:

iso-8859-1, ansi, 8bit  
dumb, ascii, none, us  
tex  
ibm7bit, 7bit, commodore, c64, digicom  
roman8  
ibmpc, pc, at, xt  
atari,  
binary, image

**/DEST**

Alle Pingpong-Hosts, die miteinander verbunden sind, werden aufgelistet. Die Zahlen zeigen die Antwortzeiten in Sekunden.

**/DEST <call>**

Fragt den Weg zum <call> ab und zeigt dabei die Übertragungszeiten ab.

**/EXCL**

Dieses Kommando ist das Gegenteil des /MSG-Befehls. Hiermit sendest Du Text an alle User dieses Kanals außer dem einen als ersten Parameter angegebenen. Da der Text intern als privater Text an die anderen verschickt wird, werden die Links etwas mehr belastet

**/FILT**

Wenn Du die Texte bestimmter User nicht lesen möchtest, so kannst Du sie hiermit in eine Liste einfügen. Alle Texte werden dann ausgefiltert, bei persönlichen Texten („/msg“) wird eine Rückmeldung an den Absender geschickt. Das Setzen/Löschen geschieht wie bei „/notify“, also z.B. „/filter + dc1ik - db4ut“ setzt dc1ik und löscht db4ut aus der Liste.

**/HELP**

Das Hilfefkommando kann von zusätzlichen Parametern gefolgt sein. Der Schrägstrich darf hier nicht vor dem fraglichen Kommando stehen, z.B.: /Help Invi. ALLE Hilfstexte können auch außerhalb des Conversmode mit „Help conversd“ als komplette Übersicht gelesen werden.

**/INVI**

Es wird eine Einladung zum genannten User geschickt. Diese Einladung wird durch das gesamte Netz geleitet. Wenn derjenige auf einem anderen Kanal ist und Dein Kanal als privat eingerichtet ist, so kann er auf Deinen Privatkanal wechseln. Wenn er im Befehlsinterpreter eines Knotens ist, so empfängt er die Einladung, er kann dann aber nicht direkt auf Deinen Privatkanal kommen, weshalb er nochmals einzuladen ist.

**/JOIN**

Verbindet Dich zusätzlich mit dem gewünschten Kanal. Im Gegensatz zu älteren Conversd-Implementationen, verbleibt man auch noch im vorherigen Kanal, denn es wird eine Mehrfach-Kanal-Verbindung unterstützt. Um einen Kanal zu verlassen, mußt Du „/leave“ verwenden. Ohne Angabe eines Kanals werden Infos zu den von Dir benutzten Kanälen ausgegeben.

**/LEAV**

Mit diesem Befehl kannst Du entweder den derzeitigen oder den angegebenen Kanal verlassen. Wenn dieser der letzte ist, so wird conversd verlassen.

**/LINK**

Der momentane Linkstatus wird angezeigt. Dies sind normalerweise Hostname, Linkstatus, Laufzeiten, Versionskodes und Statuszeit, gefolgt von der Zeit des nächsten Connectversuches und Anzahl der Versuche (auf Disconnecteten oder im Aufbau befindlichen Links), bei bestehender Verbindung werden die Queue-Längen und Bytestatistiken angezeigt.

Wenn Du Sysop bist, kannst Du Verbindungen setzen oder löschen. Es wird dann auch noch zusätzlich in Klammern der Verbindungsweg angezeigt. Syntax: /l [[-] Host [Port [via]]]

**/LIST**

Alle Kanäle, ihre Themen, Optionen und User werden angezeigt. Die Klammerwerte bedeuten:

(@) = Channel Oparator,  
(G) = Mit AWAY abwesend gemeldet,  
(!) = User ist im Sysopmodus.

**/ME**

Dieser Befehl dient dazu, den Usern auf Deinem Kanal eine Tätigkeit anzuzeigen. Wenn du z.B. „/me gähnt“ eingibst, bekommen alle User dieses Kanals folgendes angezeigt:

\*\*\* dc6iq gähnt

**/MODE**

Das Modekommando ist eines der kompliziertesten. Es wird wie folgt benutzt:

„/mo [<Kanal>] <+ ; -> <t ; i ; s ; m ; p ; l ; o <User>>“.

Die Optionen bedeuten folgendes:

t = Das Thema des Kanals läßt sich NUR von Kanal-Sysops ändern.  
i = Der Kanal wird Usern anderer Kanäle verheimlicht.  
s = Der Kanal ist geheim, die Kanalnummer wird nicht mehr angezeigt.  
m = Der Kanal ist moderiert, nur Kanal-Sysops dürfen schreiben.  
p = Der Kanal ist privat, man benötigt eine Einladung zum Einloggen.  
l = Der Kanal ist lokal, Texte werden nicht weiter verteilt.  
o <User> =Macht <User> zum Kanal-Sysop (kein - möglich).

Das Plus setzt eine Option, der Strich löscht sie. Es sind Kombinationen erlaubt, so würde z.B. „/mode 69 -s+todc6iq“ folgendes bewirken: Kanal 69 ist nicht mehr geheim, aber die Themen dürfen nur vom Kanal-Sysop gesetzt werden. Zusätzlich wird dc6iq ein Kanal-Sysop.

**/MSG**

Sendet einen Text an einen speziellen User oder an einen verbundenem Kanal. Wenn der Text an einen Kanal gehen soll, so muß man folgendes eingeben:

„/msg #<Kanal> <text>“.

Wenn das Ziel ein User ist, so kann er den Text an den zusätzlichen Sternchen erkennen. Z.B. wenn dc6iq eine Nachricht an dc1ik mit „/m dc1ik Das ist ein Test“ schickt, so erhält dc1ik folgendes: „,<\*dc6iq\*>: Das ist ein Test“.

**/NOTI**

Du wirst informiert, wenn eine bestimmte Person in der Personenliste im Convers erscheint. Z.B. fügt „/notify + dc1ik“ dc1ik in die Liste ein, „/notify - db4ut“ entfernt db4ut aus der Liste. Das Einfügen/Löschen mehrerer Calls in einem Kommando ist möglich, z.B. bewirkt „/notify + dc1ik db4ut - dc6iq dh2paf +dg3kcr“, daß dc1ik, db4ut und dg3kcr eingefügt werden sowie dc6iq und dh2paf entfernt werden. Das Entfernen von Calls, die nicht in der Liste stehen, wird ignoriert.

**/PERS**

Es kann eine kurze Beschreibung zu Deiner Person gesetzt werden, den die anderen User mit „/who“ sehen können. Z.B: „/pers Fred, Buechig, JN49fb“.

Ohne Text wird die Beschreibung gelöscht. Diese Implementation merkt sich bis zu 118 Zeichen der Beschreibung und setzt diese dann automatisch beim Einloggen (die „/char“ und „/width“ Einstellungen werden dann auch gespeichert und beim Einloggen gesetzt).

**/PROM**

Das Prompt-Kommando nimmt vier Argumente in einer zusammenhängenden Zeichenkette. „/prompt abcd“ setzt



- a = Als „/query“-Prompt.
- b = Für den normalen Prompt.
- d = ist ein Zeichen, um den Prompt zu löschen (normalerweise Backspace)
- c = Ist ein Zeichen, welches vor jedem Text, den Du empfangst, gesendet wird (normalerweise also CTRL-G).

**/QUER**

Der angegebene User ist in Zukunft der einzige Empfänger für alle Texte, die Du eingibst. Diese werden dann als private Texte an den User geschickt, wie bei „/m“. Zum Ausschalten ohne Argument aufrufen, danach geht alles wieder wie gewohnt an den Kanal. Sozusagen ein Privatmodus.

**/QUIT**

Wenn Du das eingibst, verläßt Du diesen wunderbaren Ping-Pong-Convers. Ich hoffe, es gefiel Dir.

**/TOPI**

Hiermit kann für den Kanal ein Thema gesetzt werden. Die anderen User können dieses sehen, wenn sie „/who quick“ oder „/list“ eingeben. Wenn keine Kanalnummer angegeben wird, so wird das Thema des aktiven Kanals gesetzt. Wird eine Nummer angegeben, so muß Du auch auf diesem Kanal eingeloggt sein. Um das Thema zu löschen, ist als Thema ein „,@“ einzusetzen.

**/UPTI**

Dieser Befehl zeigt an, wie lange conversd schon aktiv ist.

**/VERB**

Schaltet die Laber-Option ein/aus. Du bekommst dann viele Informationen über Aktionen der User (Einloggen/Ausloggen/Texte setzen/...), auch wenn diese nicht auf Deinem Kanal sind.

**/VERS**

Zeigt etwas Text zu dieser Version (in Englisch).

**/WHO**

Dieser Befehl hat 4 Optionen.

- a = Zeigt alle User und ihre Abwesenheitstexte, wenn gesetzt.
- l = Generiert eine LANGE Liste mit Personenbeschreibung, Abwesenheitstexte und Queue-Informationen.
- q = Gibt eine kurze Auflistung aus
- \* = Zeigt alle User Deines Kanals.

Wenn Du Informationen über bestimmte User brauchst, kannst Du die „/who u Userliste“ Variante benutzen.

**/WIDT**

Macht conversd Deine Bildschirmbreite (Zeichen/Zeile) bekannt. Die Meldungen der anderen wird dann auf diese Breite gebracht. Voreingestellt ist 80. Die Einstellung bei „/pers“ gespeichert (siehe dort).

**(!) <TheNetNode-Befehl>:**

Mit einen vorangestelltem ! ist es möglich, die TheNetNode-Befehle auch vom Convers aus aufzurufen.

**(CQ) <text>**

DER BEFEHL CQ WIRD DERZEIT VON TNN NICHT (BZW: NUR HALB) UNTERSTUETZT!!! Es wird nur noch ein UI-Frame mit dem <text> als Inhalt auf dem mit dem Parameter DefDownport eingestellten Port gesendet. Anschließend ist man wieder im Befehlsmodus von TNN.

**(D)EST**

Zeigt die Einträge der NODES-Liste in der bei den RMNC üblichen Weise an.

Destinations (5):									
DB0BID	0-0	DB0EAM	3-3	DB0EAM	4-4	DB0NHM	0-0	DB0NHM	4-4
^1.)		^2.)							

- 1.) Erreichbare Ziele (Destinations),
- 2.) SSID-Bereich des Calls.

**(D)EST <\*>**

Liste wie oben jedoch zusätzlich mit Laufzeiten.

**(D)EST <call\*>**

Hierbei muß das Call nicht vollständig sein. Die Eingabe von (D)est HB9\* zur Anzeige aller HB9.. Destinations. Sie werden mit Call, SSID-Bereich, Laufzeit und Port, auf dem sie erreichbar sind, angezeigt.

**(D)EST <call>**

Zeigt die Liste wie bei **(N)odes <call>** an .... eben nur für die FlexNet Liebhaber.

**(D)EST < <Nachbar Call>**

Zeigt die Nodes / Destinations an, die von diesem Nachbarn mit der besten Laufzeit gemeldet wurden.

**(DX)CLUSTER**

Wurde ein lokales Cluster eingetragen, so reicht die Eingabe von <dx> zum Connect dieses Clusters. Sinn dieses Befehles ist es, nicht immer auf einem Knoten nach dem nächstgelegenen Cluster suchen zu müssen.

**(G)RAPH**

Graphische Ausgabe einiger statistischer Werte. Geführt werden die Werte:

**(B)aud**

**(C)ircuits**

**(F)ree buffers**

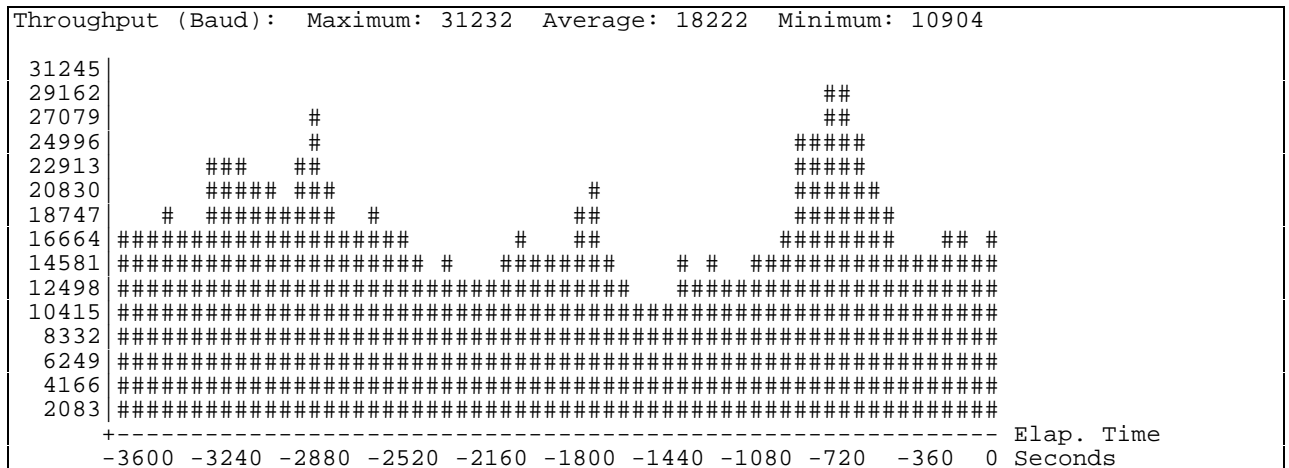
**(L)2-Links**

**(N)odes**

**(R)ounds**

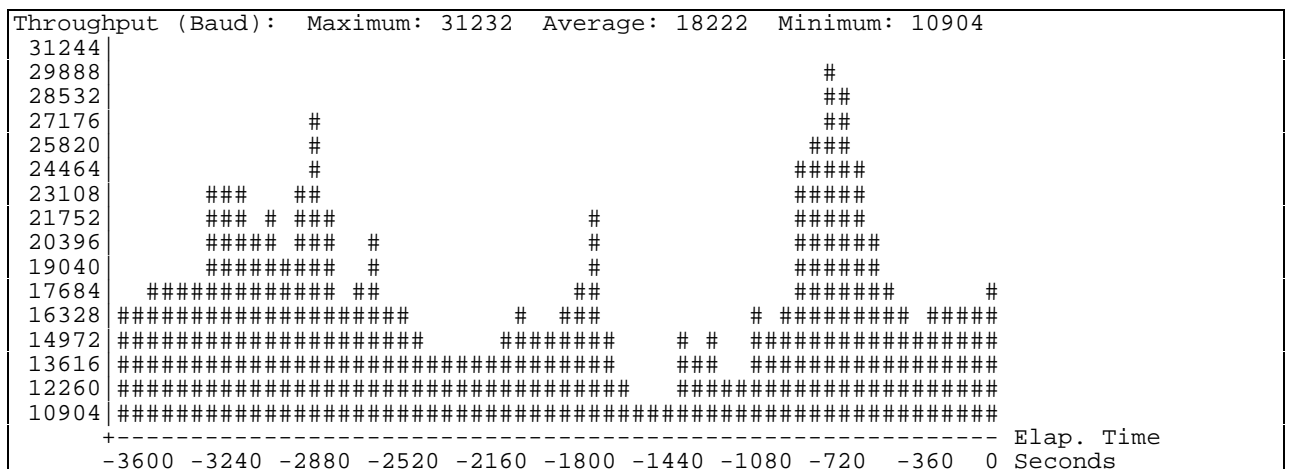
**(\*)** Für die Ausgabe aller Statistiken.

Beispiel: **(G)RAPH (B)AUD** zeigt die Entwicklung der letzten Stunde mit dem Bezugspunkt 0 auf der y-Achse an:



Um die Änderungen besser sehen zu können, kann der Bezugspunkt durch ein nachfolgendes „+“ auf den MINIMUM-Wert verschoben werden.

Beispiel: **(G)RAPH (B)AUD + .**



Wird nun noch ein **(D)AY** eingefügt, so erhält man die Ausgabe der vergangenen 23 1/2 Stunden.

Beispiel: **(G)RAPH (D)AY (B)AUD +**

**(H)ELP**

Zu den Befehlen in TheNetNode gibt es jeweils auch eine Erklärung bzw. Hilfe. Mit der Eingabe **(H)ELP** wird eine Übersicht über die möglichen Hilfen ausgegeben sowie auch die Anzahl der Bildschirmseiten. Mit **(H)ELP (N)ODES** werden die ersten 22 Zeilen ausgegeben. Die Zeilen 23-44 kommen nach der Eingabe **(H)ELP (N)ODES 2**. Wer zurückblättern kann, kann auch alle Seiten auf einmal übermittelt bekommen durch das anhängen eines „\*“. Beispiel: **(H)ELP (N)ODES \***

Programm-Version 1.15 vom 26. Dezember 1996 by DF7BZ			
TNN-Doku-Version 1.76 vom 16. November. 1998 by NORD><LINK e.V. (DG9FU)			
Folgende Befehle sind laut Dokumentation verfügbar :			
Befehl	Hilfe-Seiten		Befehl
(A)KTUELL	(extern)	1	(BE)ACON
(C)ONNECT		3	(CONV)ers
(CQ)		4	(D)EST
(DX)CLUSTER		1	(H)ELP
(HA)RDWARE	(extern)	1	(I)NFO
(L)INKS		1	(L3)MHEARD
(M)AILBOX		1	(MAP)
(MH)EARD		5	(MSG)
(N)ODES		7	(P)ARAMETER
(PAC)SAT		1	(PI)NG
(PO)RT		8	(Q)UIT
(QTH)	(extern)	2	(R)OUTES
(S)TAT		19	(SAT)
(SAV)ecall	(extern)	1	(SH)owcall
(SO)FTWARE	(extern)	1	(TA)LK
(TI)ME		1	(TOP)
(U)SER		14	(V)ERSION
Externe Befehle sind nicht bei jedem Digi vorhanden !			

**(L3)MHEARD**

Gibt eine aktuelle Rufzeichenliste der letzten 10 gehörten L3-Calls mit Datum, Uhrzeit, Port-Name, RX-Byte, TX-Byte, L3 Frame von Call und L3 Frame an Nachbar geroutet. Die L3MH-Liste wird seit der Version TNN175ag10 wie die Statistik im aktuellen Laufwerk gesichert. Sie dient dazu, die L3-Verbindungen die über den Knoten laufen, beurteilen zu können. Die SSID wird hierbei beachtet.

```
KS:DB0EAM> MHEARD (97/500)
30.01.98 17:29 P 6 [ 6095559, 12686304] DB0GOE DB0GOE
30.01.98 17:29 P 6 [ 67218, 153548] DB0MAR DB0GOE
30.01.98 17:29 P 8 [ 421886, 15142753] DB0LIP DB0LIP
30.01.98 17:29 P11 [ 5535466, 17842422] DB0KH DB0KH
30.01.98 17:29 P 7 [ 24644, 1217402] DB0AX-1 DB0AX-1
30.01.98 17:29 P 6 [ 62988, 519602] DB0PDF DB0GOE
```

MHEARD (<anzahl>/<anzahl>)

Die beiden Ziffern in der ersten Zeile geben an:

1. <anzahl> = Länge der geführten MH-Liste,
2. <anzahl> = Eingestellte Länge.

**(L3)MHEARD <anzahl>**

Gibt eine aktuelle Rufzeichenliste der letzten <anzahl> gehörten Calls mit Datum, Uhrzeit, RX-Byte, TX-Byte, L3 Frame von Call und L3 Frame an Nachbar geroutet aus.

**(L3)MHEARD <call>**

Listet wann und unter welcher SSID der Knoten mit dem <call> zuletzt ein L3-Frame über diesen Digi gesendet hat. Weiterhin werden die RX-Byte und TX-Byte (aus der Sicht des Knotens) mit angezeigt, die seit dem letzten Löschen der L3MHEARD-Liste oder Verändern der Anzahl der Listeneinträge aufgelaufen sind.

Im <call> können auch Wildcards verwendet werden. Dabei steht „\*“ für beliebig viele (oder keine) Zeichen und „?“ steht für genau 1 Zeichen.

- (L3)MHEARD df6ln = Einträge mit dem Rufzeichen DF6LN
- (L3)MHEARD df\* = Einträge von DF-Stationen
- (L3)MHEARD \*b? = Stationen mit „B“ als vorletztem Buchstaben
- (L3)MHEARD \*b\* = Stationen mit „B“ im Rufzeichen

**(L)INKS**

Zeigt die eingetragenen Rufzeichen, die den Links zugeordnet sind.

Type	Port	ALIAS	CALL	Route
L+	4	KSDXC	DB0EAM-4	
L+	5	KSBOX	DB0EAM-3	
N	6	GOE7	DB0GOE	
F	9	BID	DB0BID	
N	9	SARTG	HB9AK	via DB0BID

Typ:

- L = Localer Eintrag
- L+ = Localer Eintrag wird gemessen
- F = FlexNet-Nachbar
- N = NetRom-Nachbar

**(LOOP)**

LOOP DETECTED ist eine Warnung, daß der gewählte Verbindungsaufbau eine Schleife (Loop) bildet. Es wird also eine Interlink-Verbindung in beiden Richtungen und somit unnötig belegt. Im Gegensatz zu anderer Knotensoftware, erfolgt jedoch nur eine WARNUNG, die Verbindung kann trotzdem aufgebaut werden.

Letzteres ist aber nicht immer sinnvoll und man behindert sich selbst.. Bitte mit 2 mal Quit zum vorletzten Digi zurück gehen und dann die Verbindung direkt aufbauen... Es kann aber auch vorkommen, daß der Router von TNN einen neuen Weg zum Zielknoten gefunden hat, der zu der Schleife führt — dann kann man die Warnung auch einfach ignorieren.

**(M)AILBOX**

Wurde eine locale Mailbox eingetragen, so reicht die Eingabe eines <m> zum Connect dieser Mailbox. Sinn dieses Befehles ist es, nicht immer auf einem Knoten nach der nächstgelegenen Mailbox suchen zu müssen.

**(MH)EARD <erweiterung>**

Gibt eine aktuelle Rufzeichenliste der letzten 10 gehörten Calls mit Datum, Uhrzeit, Port-Name, RX-Byte und TX-Byte aus. Die MH-Liste wird seit der Version TNN14919 wie die Statistik im aktuellen Laufwerk gesichert. Sie dient nun auch dazu, einen User auf dem Port zu connecten, auf dem er zuletzt gehört wurde. Die SSID wird hierbei beachtet. Es ist also möglich, mit dem Call DB0XY-1 auf dem Port 0 und mit dem Call DB0XY-2 auf einem anderen Port QRV zu sein. Wer selten QRV ist, fällt nun, je nach Länge der Liste, irgendwann aus ihr raus.

<erweiterung>

Als Erweiterung kann ein "+" eingegeben werden. Es wird dann eine erweiterte User-Statistik ausgegeben. Sie besteht aus den vom User empfangenen Rej = (r) an den User gesendete Rej = (t) sowie die Anzahl der DAMA-Verstöße = (d).

```
KS:DB0EAM> MHEARD (400/500)
19.03.98 13:34 P 9 [ 927365, 1932892] DG9FU      126r   119t   0d
19.03.98 13:33 P 0 [ 59149, 407445] DG2ACM      45r    28t   0d
19.03.98 13:33 P 0 [ 24821, 856196] DK7VW      42r    23t   0d
19.03.98 13:33 P 0 [ 40804, 231704] DG2ACM-1   49r    44t   0d
19.03.98 13:33 P 0 [ 880, 43047] DK3ZL      4r     0t   0d
DG9FU de DB0EAM >
```

MHEARD (<anzahl>/<anzahl>)

Die beiden Ziffern in der ersten Zeile geben an:

1. <anzahl> = Anzahl der geführten Calls in der MH-Liste,
2. <anzahl> = Länge der MH-Liste.

**(MH)EARD <anzahl> <erweiterung>**

Gibt eine aktuelle Rufzeichenliste der letzten <anzahl> gehörten Calls mit Datum, Uhrzeit, RX-Byte und TX-Byte aus.

**(MH)EARD <call> <erweiterung>**

Listet wann und unter welcher SSID der User mit dem <call> den Knoten zuletzt benutzt hat. Weiterhin werden die RX-Byte und TX-Byte (aus der Sicht des Knotens) mit angezeigt, die seit dem letzten Löschen der MHEARD-Liste oder Verändern der Anzahl der Listeneinträge aufgelaufen sind.

Im <call> können auch Wildcards verwendet werden. Dabei steht "\*" für beliebig viele (oder keine) Zeichen und "?" steht für genau 1 Zeichen.

- (MH)EARD df6ln = Einträge mit dem Rufzeichen DF6LN
- (MH)EARD df\* = Einträge von DF-Stationen
- (MH)EARD \*b? = Stationen mit "B" als vorletztem Buchstaben
- (MH)EARD \*b\* = Stationen mit "B" im Rufzeichen

## (N)ODES

Zeigt alle momentan bekannten Knoten an, die das NET/ROM- bzw. TheNet-L3-Protokoll verwenden. Die ausgegebene Liste wird regelmäßig über sogenannte Rundspruchsendungen (Broadcast) der TheNet-Nachbarknoten auf dem neuesten Stand gehalten. Die Liste ändert sich also, wenn Links ausfallen oder neue Links hinzukommen. Außerdem ändern sich die Laufzeiten der einzelnen Einträge in Abhängigkeit von der Linkbelastung und bei schlechten Ausbreitungsbedingungen. Beispiel:

1	2	3	4
/	/	/	/
KS:DB0EAM > Nodes (139/1009):			
SH9600:DB0AZ	HUSUM:DB0HES	HHWEST:DB0HHW	HL:DB0MAR
KIELMB:DB0OQ	SL:DB0SUE	SYF7:OZ3DIJ-7	
/	/		
5	6		

- 1.) Alias dieses Netzknoten.
- 2.) Rufzeichen dieses Netzknoten.
- 3.) Anzahl der bekannten Knoteneinträge.
- 4.) Maximal mögliche Anzahl an Einträgen
- 5.) Alias des Endknoten.
- 6.) Rufzeichen des Endknoten.

ALIAS ist dabei eine maximal 6-stellige Abkürzung zur besseren geographischen Einordnung des Knotenstandortes. In unserem Raum werden normalerweise die üblichen Autokennzeichen als ALIAS verwendet oder aber kurze Ortsnamen auch ausgeschrieben (wie z.B. bei KS:DB0EAM oder KIEL:DB0IL). Zu den in der Liste aufgeführten Endknoten kann in der Regel mit dem Connect-Befehl eine Verbindung hergestellt werden. Endknoten, deren ALIAS mit „BOX“ oder „MB“ endet, sind damit als Mailbox erkennbar; DX-Cluster sind mit „DX“ oder „DXC“ am Ende des ALIAS erkennbar.

Die Anzahl der bekannten Netzknoten wird hinter „Nodes“ in Klammern angezeigt.

Der Nodes-Befehl kann bis auf den ersten Buchstaben abgekürzt werden und auch beliebig groß und / oder klein geschrieben werden. Der Nodes-Befehl kann außerdem mit Parametern aufgerufen werden, um Informationen zu einzelnen Endknoten oder Gruppen von Endknoten zu bekommen.

Mit dem Befehl:

(N)ODES <call> oder (N)ODES <alias>

bekommt man eine Auflistung der bekannten Wege zu dem mit <call> bzw. <alias> angegebenen Endknoten. So erhält man z.B. mit „(N)odes DB0FC“. Weiterhin wird hiermit der NetRom - Route - Rekord (NRR) ausgelöst.

(N)ODES <call> \* oder (N)ODES <alias> \*

Ein zusätzliches „\*“ hinter <call> zeigt alle Wege an.

		1		2															
		/		/															
Routes to		BRO:DB0BRO																	
---T[ms]---		RxT---		TxT---		LT-Mode-		Obc----		RTT-Po-		Route-----							
>	4330	3910	0	2	DG	0		420	6	DB0GOE									
	16440	7000	4870	10	DG	19		9440	9	HB9AK via DB0BID									
	30970	17560	4870	5	DG	0		13410	9	DB0LEB-7 via DB0BID									
	600000	599990	4870	4	DG	0		1550	9	DB0KH via DB0BID									
>	4800	4800	0	10	VC	0		910	10	DB0NHM									
	5970	10	0	1	DG	0		5960	10	DB0BRO via DB0NHM									
	25200	3480	0	2	DG	0		21720	10	DB0GOE via DB0NHM									
	600000	599990	4870	4	DG	0		520	11	DB0KH									
^	/	/	/	/	/	/		/	/	/									
3	4	5	6	7	8	9		10	11	12									

- 1.) ALIAS-Kennzeichen des gefragten Endknoten.
- 2.) Rufzeichen des gefragten Endknoten.
- 3.) > = Der Weg wird derzeit verwendet für eine Verbindung zu dem Endknoten.  
+ = Zeigt eine alternative Route an, über die auch Datenübertragung stattfindet.  
- = Diese Route ist abgemeldet und wird nicht mehr benutzt.  
\* = Es wird kein Obs mehr geführt, da der Nachbar bereits differenziellen Broadcast unterstützt.
- 4.) Gemessene Laufzeit in ms des jeweiligen Weges.
- 5.) RxT Laufzeit die vom dem Nachbar über das Ziel DB0BRO gemeldet wurde.  
Bzw. 10 wenn es ein direkter Nachbar ist.
- 6.) Laufzeit mit der das Ziel weitergemeldet wird. (Errechnet sich aus RxT und gemessener Laufzeit zum Nachbarknoten).
- 7.) Lifetime des Knotens. Wenn LT=0 wird der Knoten nicht mehr weiter verbreitet.
- 8.) Übertragungsmodus auf diesem Link DG = Datagramm (Höhere Protokollebene die das Umrouten ermöglicht)  
; VC = Virtual Circuit (Unterste Protokollebene).
- 9.) Obc Obsolentcounter (Veraltenszähler) für Ziele mit altem Protokoll.
- 10.) Round - Trip - Timer oder Laufzeit zum Nachbarknoten.
- 11.) Port über den die Verbindung geroutet wird.
- 12.) Rufzeichen des Nachbarknotens für den jeweiligen Weg.

Bei einer Laufzeit 600000 ms wurde der Weg über Fastlearn aufgenommen.

NetRom - Route - Rekord (NRR) oder die Ermittlung des Übertragungsweges zwischen 2 Knoten die dieses Protokoll unterstützen. Es wird an das Node ein NRR - Frame gesendet. Auf dem Hin- und Rückweg hängen alle Knoten Ihr Call in dieses Frame ein. Der Knoten, der das Frame an den Absender zurücksendet kennzeichnet sich selbst mit einem „\*“. Ein „DB0EAM\*<flexgate>“ oder „DB0EAM\*<local>“ bedeutet, dass dieser Digipeater das abgefragte Node über ein FlexGate oder einen local eingetragenen Link auf Level2 umsetzt. Digipeater die NRR nicht unterstützen, jedoch Level3-Frames transportieren, erscheinen in der Liste mit einem „?“ . Unterstützt das abgefragte Node jedoch NRR nicht, so kommt keine Antwort zurück.

KS:DB0EAM> Route (DG): DB0EAM DB0GOE DB0BRO\* DB0EAM

#### Folgende Konstellation bei DB0SHG:

Routes to		BS:DB0FC																	
---T[ms]---		RxT---		TxT---		LT-Mode-		Obc----		RTT-Po-		Route-----							
-	33800	33800	33900	10	VC	0		12880	4	DB0HSK									
>	17700	17700	0	10	VC	0		77830	6	DB0HW									
	43360	10	0	1	DG	21		43350	6	DB0FC via DB0HW									

bedeutet nichts anders, als das ein User der DB0FC connecten möchte, den FlexNet-Weg über DB0HW vorgeschlagen bekommt, weil sonst keine wesentlich schnellere Alternative zur Verfügung steht. Eine solche Verbindung kann natürlich NICHT umgeroutet werden, selbst wenn zwischendurch ein besserer NETROM-Weg zur Verfügung steht !

Umgedreht wird nie eine NETROM-QSO auf FlexNet-Weg umgeroutet.

Wie war es früher (1.70)....

Da gab es zwei Listen, eine NETROM und eine FlexNet. Es wurde immer zuerst der NETROM-Weg genommen, egal wie schlecht er war ! Im schlimmsten Fall bedeutete dies, NETROM 10 Minuten Laufzeit, FlexNet-Weg 10 Sekunden. Dieses war jedoch wenig sinnvoll. Heute gibt es nur noch eine Liste, dafür werden FlexNet-Ziele mit einigen % unterbewertet. Sind mehrere NETROM-Wege vorhanden so wird natürlich weiterhin umgeroutet.

Hin- und Rückweg können bekanntlich unterschiedlich Wege nehmen.

Um erweiterte Informationen über eine Gruppe von Endknoten zu bekommen, gibt man den Befehl ein:

(N)ODES <laufzeit> <name>

Für <laufzeit> wird dabei der Wert für die minimale Laufzeit der Endknoten angegeben. Soll nicht eine minimale, sondern eine maximale Laufzeit angegeben werden, wird nach der Laufzeit ein Minus geschrieben. Für <name> kann ein Rufzeichen oder ein ALIAS eingesetzt werden, bei dem „\*“ und „?“ als Platzhalter verwendet werden können. Dabei steht „\*“ für beliebig viele (oder keine) Zeichen und „?“ steht für genau 1 Zeichen. Außerdem kann man die Abfrage auf den ALIAS-Teil oder auf den Rufzeichen-Teil des Knoteneintrages beschränken, indem man ein „:“ vor oder hinter <name> setzt. Mit „<name>:“ begrenzt man die Abfrage auf den ALIAS-Teil, und mit „:<name>“ begrenzt man die Abfrage auf den Rufzeichen-Teil der Knoteneinträge. Wird nur eine <laufzeit> angegeben, so werden alle Knoten mit passender Laufzeit angezeigt. Wird nur ein <name> angegeben, so wird die Laufzeit bei der Auswahl nicht berücksichtigt. Die ausgegebene Liste an Endknoten sieht dann z.B. so aus:

```
KS:DB0EAM > Nodes: (7)

HHS:DB0HBS 176/11 HHSBOX:DB0HBS-1 69/11
/         /         /         /
1         2         3         4
```

- 1.) ALIAS des Endknoten.
- 2.) Rufzeichen des Endknoten.
- 3.) Laufzeit in 10 ms Schritten für den besten bekannten Weg.
- 4.) Port für Nachbarknoten.

#### Beispiele:

```
Befehl      ! Antwort
=====
NODES 2000   ! Alle Einträge mit Laufzeiten min. 2000.
NODES -60    ! Alle Einträge mit Laufzeiten max. 60.
NODES 100 d*  ! Rufzeichen oder ALIAS beginnt mit „D“, Laufzeit min. 100.
NODES -100 *d ! Rufzeichen oder ALIAS endet mit „D“, Laufzeit max. 100.
NODES *d*     ! Rufzeichen oder ALIAS enthält ein „D“, Laufzeit egal.
NODES ?????? ! Rufzeichen oder ALIAS ist genau 5 Zeichen lang.
NODES :d*     ! Rufzeichen beginnt mit „D“.
NODES :?b*    ! Rufzeichen mit „B“ als 2. Buchstaben (DB- u. HB9-Stationen).
NODES :hb9*   ! HB9-Rufzeichen.
NODES :*l     ! Rufzeichen endet mit „L“.
NODES k*:     ! ALIAS beginnt mit „K“.
NODES *box*:  ! ALIAS enthält „BOX“.
NODES *dx:    ! ALIAS endet mit „DX“.
NODES *u?:    ! ALIAS mit „U“ als vorletztem Buchstabe.
NODES *       ! Zeigt die gesamte Liste an.
```

(N)ODES <<Nachbar Call>

Zeigt die Nodes / Destinations an, die von diesem Nachbarn mit der besten Laufzeit gemeldet wurden.

## (P)ARAMETER

Ausgabe der Parameter-Liste.

KS:DB0EAM> Parms:					
01:NoAckBuf	20	02:Timeout	3600	03:SaveConfig	3
04:DAMA-Speedf	4800	05:DAMA-MaxPri	10	06:DAMA-MaxPol	5
07:DAMA-Tout	50	08:CommandLog	0	09:SysopLog	1
10:Downport	15	11:TestSSID	15	12:ConvSSID	1
DG9FU de DB0EAM (13:12)>					

## 01: NoAckBuf

Überfüllungsgrenzwert in Anzahl Frames. Anzahl der Pakete, die auf der Transport-Layer-Ebene zwischengespeichert werden, bis eine Choke-Nachricht zum vermittelten Knoten geschickt wird. Gleichzeitig die Anzahl Frames, die im Link-Layer zwischengespeichert werden, bevor das Link-Layer in den Busy-Zustand geht. Dieser Grenzwert\*\* verhindert den Überlauf eines TheNet-Knoten, falls über das Transport-Layer zu viele Pakete einlaufen, oder falls eine Station in einem Link zu viele Pakete auf einmal senden will.

## 02: Timeout

Keine-Aktivität-Timeout in Sekunden, (0 = abgeschaltet). Passiert auf einem Layer-2-Link oder Layer-4-Circuit solange kein Informationsaustausch, wird der Link abgebrochen.

## 03: SaveConfig

Zeitraum in 10 min-Schritten, in dem TNN176.STA und MHEARD.TAB mit den aktuellen Statistikdaten auf Disk oder HD gespeichert wird.

## 04: Dama-Speedf

Bei TNN-Digis mit Multibaud (z.B. 1200 und 9600 Baud) bestand ebenfalls ein weiterer Wunsch, einen Anreiz für 9k6-Betrieb zu schaffen. Im Normalfall werden 1200- und 9k6-User gleichbehandelt, und man kann lediglich über MAXFRAME den 9k6-Betrieb etwas bevorzugen. Üblicherweise sollte man MAXFRAME 7 auf 9k6 benutzen und nicht mehr als MAXFRAME 2 auf 1200 Baud. Aber einigen Sysop war dies noch nicht genug. Was der neue DAMA-Speedfaktor bewirkt, soll nun am folgenden Beispiel erläutert werden. Angenommen, wir haben einen 1200-Baud- und einen 9600-Baud-Einstieg, und wir wollen die 9600er User deutlich bevorzugen. Steht DAMA-Speedf jetzt auf 9600, dann werden User auf einem 1200-Baud erst in jeder 8. Runde bedient. Auf den ersten Blick scheint dies viel zu langsam, aber bei 9k6 sind die Frames ja so kurz, so daß es also NICHT gleich um Faktor 8 langsamer geht. Sind überhaupt keine 9k6-User da, dann ist alles wie bisher. Wichtig: Die korrekte Baudrate muß im Speed-Parameter für den jeweiligen Port richtig eingetragen sein.

Die „Verlangsamung“ berechnet sich so:

		DAMA-Speedf	9600		
Verlangsamung	=	-----	=	-----	= 8
		Speed[Port]	1200		==

Bei dem 9k6-Port ergibt sich entsprechend ein Faktor von 1. Hat man gleichzeitig noch einen 2k4-Einstieg, dann ist die Verlangsamung dort  $9600/2400 = 4$ . Möchte man den 1200er Einstieg nicht gleich um Faktor 8 beeinflussen, kann man natürlich einen beliebigen Wert zwischen 1200-9600 (oder mehr!) bei diesem Parameter eintragen. Ein Wert kleiner als die niedrigste Baudrate bzw. ein Wert von 0 schaltet diese Funktion ab.

## 05: DAMA-MaxPri

Maximaler Zählerstand, der erreicht werden kann bei geringer Aktivität eines User. Dieser User wird dann seltener zu Gunsten der anderen User abgefragt, ob er Daten zu senden hat.

## 06: DAMA-MaxPol

0 = ausgeschaltet.

x = Nach x Verstößen erfolgt ein Abwurf vom DAMA-Master.

Die TNN-Software bietet für den SYSOP eine **abschaltbare** DAMA-Kontrolle. Alle User, die sich nicht an das DAMA-Protokoll halten, bekommen bei jedem DAMA-Verstoß eine entsprechende WARNUNG mitgeteilt. Ist eine, vom Sysop einstellbare, Anzahl überschritten, dann erfolgt ein Abwurf des User vom Digi! Die „Meckermeldungen“ werden nicht mehr nutzlos als UI-Frame (Bake) ausgesendet werden, sondern sie werden jetzt mitten in das betreffende QSO eingehängt. Die Warnung gibt die Anzahl der aktuellen Verstöße an sowie die Anzahl der Verstöße, bis ein Disconnect seitens des DAMA-Master erfolgt (mit entsprechender Meldung).

## 07: DAMA-Tout

Zeit in 10 ms, die gewartet wird, wenn auf ein Paket an einen User keine Bestätigung kommt, bis zum Senden an den nächsten User in der Reihenfolge.



## 08: CommandLog

Schafft nun die Möglichkeit, eine Logdatei COMMAND.LOG im Verzeichnis TNN zu führen. Es werden alle Eingaben im Knoten mit CALL, DATUM, UHRZEIT und COMMAND in der Datei abgespeichert. Es sind folgende Einstellungen möglich:

0 = Keine Protokollierung.

1 = Nur Sysop-Eingaben werden protokolliert.

2 = Alle Eingaben werden protokolliert (Vorsicht, die Logdatei wächst schnell!).

## 09: SysopLog

Speichert wer sich als SYSOP beim Knoten privilegiert.

## 10: Downport

Legt den Port fest, auf dem die Connect ausgesendet werden die der Knoten nicht zuordnen kann. Z.B.: Wenn ein Knoten connected werden der nicht in der NODES-Liste steht oder ein User connected werden soll der nicht in der MH-Liste steht. Sollen die nicht zuzuordnenden Connect nicht auf einem User-Einstieg ausgesendet werden, so kann auch ein ausgeschalteter Port zugewiesen werden. Die connectende Station bekommt dann ein „Port not in use“ zurück.

## 11: TestSSID

Legt die SSID fest, mit der dieser Knoten die Laufzeitmessung durchführt bei einem mit L+ eingetragenes Call.

## 12: ConvSSID

Einstellung der SSID die der Convers benutzen soll.

## (PAC)SAT

Zeigt Call und Message-Pool des BROADCAST-Server an.

```
Server call: XX0XX-2
Message pool: 1-9 (9 Messages)
```

## (PI)NG &lt;ipadd&gt;

Führt eine Laufzeitabfrage zu der IP-Nummer durch.

## (PO)RT

Gibt eine Liste mit den aktuellen Porteeinstellungen aus: (Diese Darstellung ist ein MISCHMASCH aus DOS/GNU- und LINUX- Version und dient hier nur zur Verdeutlichung.)

```
TEST:DB0XX> Link-Interface Ports:
-#-Name-----Speed/Mode-Max-TXD-----Hardware--
0 70cm_1200      1200m      2   20  DAMA CTEXT      MH VAN v4.04E
1 70cm_9600      9600m      7   15  DAMA CTEXT      MH VAN v4.04E
2 23cm_9600      9600       7   10      CTEXT      MH VAN v4.04E
3 Lohfelden     9600d      4   10      CTEXT      TOKENRING/dev/ttyS0
4 Mailbox       64000d     7    0      CTEXT      TNN-ETH
5 HHW           9600       7    5      CTEXT SYSOP    SMACK /dev/ttyS5
6 Fyn           1200       4   25      CTEXT      SMACK /dev/ttyS3
7 SEG           9600       7    5      CTEXT      SMACK /dev/ttyS1
8 Cluster      19200      7   200     KISS /dev/ptyz3
```

#:

Gibt den Port des Knotens an.

Name:

Zur leichteren Übersicht.

Speed/Mode:

Speed zeigt die eingestellte Baudrate sowie zusätzlich gesetzte Flags an.

d : Duplex.

c : CRC bzw. DCD bei 1k2-Modem.

r : ext. Takt (rx).

t : ext. Takt (tx).

e : ext. Takt beide (Vanessa).

m : Multibaud-Kopplung (Vanessa, SCC).

z : NRZ statt NRZI.

TXD:  
eingestelltes TXDelay.

Max:  
eingestelltes MaxFrame.

DAMA:  
Auf diesem Port eingeschaltet.

CTEXT:  
Auf diesem Port werden die Connect-Texte ausgesendet.

SYSOP:  
Auf diesem Port ist der SYSOP-Mode aktiviert. Ein L2-Connect ist nur mit Paßwort möglich.

MH:  
Auf diesem Port wird die MH-Liste geführt.

Hardware:

Tokenring:	Serielle Übertragung auf dem Tokenring.
Van 4.04e:	Parallele Übertragung zum Frontendrechner Vanessa mit der Softwareversion.
Kiss1:	Übertragung über die eingestellte serielle Schnittstelle ohne CRC.
SMACK1:	Wie KISS1 jedoch mit SMACK-CRC.
RKISS1:	Wie KISS1 jedoch mit RMNC-CRC.
IPX_PD:	Hier ist eine Ethernet-Karte im Einsatz mit IPXPD-Treiber.
TNN-ETH:	Ethernet Packet Treiber.

Bei Linux wird zusätzlich die Device-Bezeichnung des Port angegeben.

#### (PO)RT \*

Gibt eine Liste mit den aktuellen Port Parametern aus:

KS:DB0EAM> Port Parameters:							
-#	Port	TXDelay	Pers	Slot	IRTT	MaxFrame	L2Retry-Timer2
0	:70cm_1200	20	255	20	1000	2	10 480
1	:70cm_9600	15	255	15	150	7	10 60
2	:23cm_9600	10	127	10	80	7	20 30
3	:Lohfelden	10	255	10	80	4	20 30
4	:Cluster	10	255	10	80	7	20 30
5	:Mailbox	10	255	10	34	7	20 7
6	:Goettingen	10	255	10	80	7	20 30

Nur TXDelay und MaxFrame werden noch vom Sysop eingestellt. Die anderen Parameter stellt TNN selbst ein.

TXDelay:  
Sendervorlaufzeit nach dem Hochtasten des Senders bis zur Aussendung des ersten Datenpaketes in 10ms.

Pers:  
Persistence Wert (0-255).

Slot:  
Zeitschlitz-Intervall (Slot time intervall) in 10ms. Dieser Parameter gibt die Dauer des Zeitschlitzes für die Persistence-Steuerung an. Jedesmal, wenn der TNC ein Paket ausstrahlen wollte und die unter Slot-Time beschriebene Zufallszahl außerhalb des Persistence-Bereichs lag, wird dann für die Dauer des Zeitschlitzes gewartet und anschließend die Persistence-Prozedur erneut durchlaufen.

IRTT:  
Bedeutet Initial-Round-Trip-Time und ist der Anfangswert für die RTT-Berechnung, also der Wert, der für die erste Berechnung anstelle von SRTT benutzt wird, wenn noch keine Messung erfolgen konnte.

Der Timer 1 (wann soll der TNN Nachfragen, wenn er Deine Antwort nicht gehört hat) berechnet sich wie folgt:

Für RETRIES kleiner gleich 3:	$T1 = SRTT * 3$
für RETRIES größer 3:	$T1 = SRTT * (RETRIES + 4)$

Beim Connecten wird SRTT aus dem IRTT berechnet (siehe Parms):

$$\text{SRTT} = (\text{DIGI} * 2 + 1) * \text{IRTT}$$

Bei einem direkten Connect entspricht SRTT dann dem IRTT, mit Anzahl DIGI wird es dann entsprechend größer. Wenn nun auf einem Ports ein IRTT = 500 = 5s aus der Baudrate errechnet wurde, bedeutet das:

Retry 1..3 →  $T1 = 5 * 3 = 15\text{s}$  bis zur Nachfrage !

z.B. bei Retry 6 →  $T1 = 5\text{s} * (6 + 4) = 50\text{s}!!!$

Bei laufendem Betrieb bewegt sich der SRTT dann dynamisch.

#### MaxFrame:

Anwender-Link MaxFrame in Anzahl der Frames. Anzahl der Infopakete auf Layer2-Ebene, die ohne Erhalt einer Bestätigung hintereinander ausgesendet werden dürfen.

#### L2Retry:

Bestimmt die Anzahl der Versuche, um auf Layer2-Ebene Kontakt zu einer anderen Station zu bekommen (Antwort auf Kommandos und Poll). Nach dieser Anzahl von Versuchen wird der Link als defekt gemeldet, falls keine Antwort erfolgt.

#### Timer2:

Anwender-Link T2 in 10ms. Dieser Timer bestimmt die Wartezeit, nachdem ein eingehendes Informationspaket bestätigt wird mit einem RR/REJ/RNR-Paket. Einerseits ist diese Verzögerung zur Durchsatzsteigerung da, weil man in diesem Intervall anderen eine Chance zum Senden gibt, andererseits wird dem Sende-Layer die Chance gegeben, eine Bestätigung in ein zu sendendes Infopaket zu packen und somit ein Link-Layer-Paket einzusparen. (Das Ergebnis ist in der Statistik abzulesen.)

#### (Q)UIT

Verbindung wird vom Knoten aufgelöst (DISCONNECT). Dadurch ergibt sich die Möglichkeit, zum vorher connecteten Knoten reconnected (zurückverbunden) zu werden. Wenn vorhanden, wird der Text QUIT.TXT vor dem Auflösen der Verbindung gesendet.

#### (R)OUTES

Zeigt die bestehenden Routen zu den Nachbarknoten an.

Routes of KS:DB0EAM (18/32)						
Node-----	Typ	Po	---Dst---	L3SRTT[ms]	---	State--Route-----
DB0EAM-3	L+	5	1	170/0		aktive
DB0VFK	L	3	1	0/0		
DB0GOE	I	6	126	400/850		conn.
DB0BID	F	9	544	450/820		conn.
DB0II	I	9	86	0/58460		conn. DB0BID
DB0KH	N+	9	102	7590/5030		conn. DB0BID
DB0KH	N+	11	121	430/710		conn.

#### Routes of KS:DB0EAM (18/32)

Die Routesliste ist auf das wesentliche gekürzt ! Die (18/32) besagt, daß in der Linkliste 18 von 32 möglichen Einträgen benutzt sind. Die Länge der Linkliste kann ggf. mit SET TNNCFG=xxxx,xx angepaßt werden. (siehe: START.BAT)

#### NODE :

Anzeige des Rufzeichens des benachbarten Knotens.

#### Type :

Gibt an, um welchen Typ es sich bei diesem Eintrag handelt. Möglich sind:

#### Type I :

Nachbar arbeitet mit dem neuen InterNode - Protokoll, in dem TheNetNode nur Laufzeiten austauscht.

#### Type L :

Local Call und Alias wird mit Laufzeit 4000 ms eingetragen. Es wird keine Laufzeitüberprüfung durchgeführt.

Wird ein L+ eingesetzt, so wird das Ziel alle 5 Minuten connected. Ist das Ziel QRV, so wird dieses mit der Laufzeit in die Routesliste eingetragen und weitergemeldet.

#### Type F :

FlexNet Port arbeitet mit FlexNet Protokoll.

Type N :

NetRom-Nachbar kann das Level-3 Protokoll alter Art, sendet jedoch die Meßframes noch unprotokolliert zurück.

Type N+ :

NetRom-Nachbar verwendet das neue Protokoll, wo unerreichbare Ziele sofort abgemeldet werden. Die Übermittlung und Messung geschieht im Level-3 nun protokolliert.

Type N- :

NetRom-Nachbar verwendet das alte Protokoll. Nicht erreichbare Ziele fallen auf Grund der Timer aus der Nodesliste. Die Übermittlung und Messung geschieht im Level-2 als UI-Frames und damit NICHT gesichert!

Po :

Port, über den die Verbindung läuft.

Dst :

Anzahl der Ziele die von dem Nachbarn gemeldet wurden.

L3SRTT[ms] :

Zeigt die zu diesem Node ermittelte Laufzeit in Millisekunden an, sowie die der Gegenseite.

State setup :

Es wird versucht eine Verbindung zum Nachbarn aufzubauen.

State conn. :

Die Verbindung zum Nachbarn ist erfolgreich aufgebaut worden und bleibt von nun an bestehen. NUR wenn diese Level2 Verbindung besteht, werden auch die NODES / DESTINATIONS von diesem Nachbarn in die NODES-/DESTINATION-Liste übernommen. Damit werden nun Zielknoten aus dem Netz entfernt, die nicht sicher zu erreichen sind oder zu denen der Link nur in einer Richtung besteht.

Route :

Digipeater, die nicht das TheNet-Protokoll benutzen bis zum eingetragenen TheNet-Knoten.

**(R)OUTES (V)ersion :**

Zeigt zusätzlich den Softwarestand der Nachbarknoten mit an und was für ein Broadcast dahin gemacht wird.

UI-Broadcast	= Alter TheNet Broadcast,
Info-Broadcast	= Broadcast wird gesichert im Level 3 übertragen,
Differential-Broadcast	= Es werden nur die Änderungen übertragen.

(S)TAT :

Gibt die Statistik, die im Knoten geführt wird, aus. Mit (S)TAT werden jedoch nur noch die Kopfdaten ausgegeben.

Weitere Statistiken:

(S)tat (*)	Ausgabe aller Statistiken;
(S)tat (E)rror	Kopf und Error-Statistik;
(S)tat (H)ardware	Kopf und Hardware-Statistik;
(S)tat (I)p	Kopf und TCPIP-Statistik;
(S)tat (L)ink	Kopf und Link-Statistik;
(S)tat (P)orts	Kopf und Port-Statistik.

#### Die Kopfdaten ....

System Statistics: 01.04.98 00:00:13 - 25.04.98 08:45:18				
Startup: 21.04.98 16:29:58				
Uptime: 3/16:15				
	(min)	(now)	(max)	
Rounds/sec:	337	737	1487	
Free Buffers:	1747	2369	8123	
Overall Throughput:		32400	58208	Baud
Active L2-Links:		109	179	
Active Circuits:		25	53	
Active Nodes:		781	3001	
CPU load: 51%				
Buffer usage: 71%				
Network Heap: 817800 Bytes				
Free RAM (Fmem): 1642496 Bytes				

System Statistics:

Datum und Uhrzeit des letzten Löschsens der Statistik sowie das Auslesedatum und - Uhrzeit. Wichtig bei automatischen Abrufen der Statistik für Diagramme.

Startup: 14.02.97 16:29:29

Letzter Neustart der Software.

Uptime = 0/16:58

Laufzeit des Programmes in Tagen/Stunden:Minuten.

Die folgenden Angaben sind jeweils in dieser Reihenfolge aufgeführt: minimaler Wert aktueller Wert maximaler Wert

Rounds/sec: <anzahl> <anzahl> <anzahl>

Gibt die Hauptschleifendurchläufe des Programmes an. Bei der DOS/GNU-Version sind die angezeigten Werte abhängig von der Rechnerperformance. Bei Linux wird als aktueller Wert üblicherweise ca. 50 angezeigt — mit Tokenring ca. 100.

Free Buffers : <anzahl> <anzahl> <anzahl>

Zeigt die freien Buffer des Knotens an.

Ein Buffer sind 64 Info-Byte und 8 Listenzeiger-Byte, insgesamt also 72 Byte. Jede zwischengespeicherte Information und auch jeder Eintrag in der MHEARD-Liste belegt Buffer. Wenn TheNetNode keinen freien Buffer mehr zur Verfügung hat, wird ein Reset ausgelöst, der alle bestehenden Verbindungen mit totalem Informationsverlust löscht. Im Normalbetrieb ist dieser Fall aber nahezu ausgeschlossen, da bei normal gesetzten TheNetNode-Parametern eigentlich nie zu wenig Buffer vorhanden sein können (man kann nicht beliebig viele Pakete hintereinander im Knoten „abladen“).

Overall Throughput : <anzahl> <anzahl> <anzahl>

Zeigt den Datendurchsatz des Knotens an. Hierbei handelt es sich um reine Infodaten ohne Protokoll- und Overheadbytes !

Aktive L2-Links : <anzahl> <anzahl>

Anzeige der Anzahl der aktiven und maximal gleichzeitig vorgekommenen Level-2-Verbindungen.

Active Nodes: <anzahl> <anzahl>

Hier wird die Anzahl der derzeit bekannten Nodes angezeigt. Die zweite Anzahl gibt die bisher maximal eingetragenen Nodes an.

Aktive Circuits : <anzahl> <anzahl>

Anzeige der Anzahl der aktiven und maximal gleichzeitig vorgekommenen Circuit-Verbindungen. Hier werden natürlich nur die Verbindungen angezeigt, die auf diesem Knoten beginnen oder enden. Alle anderen Verbindungen werden ja virtuell durch das Netz geleitet und brauchen nicht auf jedem Knoten verwaltet zu werden.

CPU load : 39%

Zeigt die Auslastung der CPU an. Hier wird das Verhältnis aus maximaler und aktueller Zahl an Rounds/sec als Maß für die CPU-Auslastung berechnet.

Buffer usage : 14%

Verhältnis zwischen belegten und zur Verfügung stehenden Buffern in %.

Network Heap: <anzahl>

Sind die durch Routing - Infos belegten Bytes.

Free RAM (Fmem) : <anzahl>

Freier Hauptspeicher für DOS. Bei der DPMI-Version der freie Gesamtspeicher (bis 16 MB). Bei der Linux-Version ist dieser Wert wenig aussagekräftig.

#### Die Hardware - Statistik....

##### Tokenring-Statistics:

Tokens/sec:	92	161	180
TOKENRING load:	11%		
Token ring speed:	38400 Bit/s.		
Token-Recoveries:	1		
Bad-Frames:	0		
TNC#1 RxAdr:	0	RxErr:	0
TNC#3 RxAdr:	0	RxErr:	0

Tokens/sec: <anzahl> <anzahl> <anzahl>

Zeigt die Aktivitäten des Tokenringes an.

TOKENRING load : 36%

Zeigt die Auslastung des Tokenring an.

Tokenring speed: 38400 Bit/s.

Anzeige der eingestellten Baudrate des Tokenringes.

Token-Recoveries: <anzahl> <datum>

Sind Abfragen an die TNC, die nicht wieder zum Rechner zurückkommen.

BAD Frames:

Defektes oder falsches Token. Diese Fehler werden auf Port 0 gezählt, da nicht festgestellt werden kann, auf welchem port der Fehler entstanden ist.

TNC#1 RxAdr: 0 RxErr: 0:

Sowie Fehler der einzelnen TNC.

##### KISS-Statistics:

KISS1 RxCRC:	0	RxErr:	0
--------------	---	--------	---

KISS1 RxCRC: 0 RxErr: 0

Fehler dieser Kiss - Schnittstelle.

RxCRC Fehler:

Ein Frame mit falscher CRC wurde auf einem SMACK- oder RMNC-Kisslink empfangen. Diese Fehler werden erkannt und sind unkritisch, weisen aber auf eine schlechte Übertragung hin.

```
External-Statistics:
TNN-ETH      TxErr: 0  RxOvr: 0  OFlow: 0  IOErr: 0
```

Fehler die durch externe Treiber entstehen.

#### Die Error - Statistik....

```
Error-Statistics:

      RxID  RxLen  RxCtl  Resets
0:70cm_1200    26      0      0      0
1:70cm_9600   347      0      0      2
2:23cm_9600   383      3      0      0
3:Lohfelden   170      3      0      2
4:Cluster      0      0      0      0
5:Mailbox      0      0      0      2
6:Goettingen   0      0      0      0
```

Resets:

Anzahl der Resets der entsprechenden Port Hardware (Vanessakarten oder TNC im Tokenring).

#### Die IP - Statistik ....

```
IP-Gateway-Statistics:

ipForwarding:      1      ipDefaultTTL:      10      ipInReceives:      0
ipInHdrErrors:     2      ipInAddrErrors:     0      ipForwDatagrams:    2
ipInUnknownProtos: 0      ipInDiscards:       0      ipInDelivers:       0
ipOutRequests:     1      ipOutDiscards:      0      ipOutNoRoutes:      2
ipReasmTimeout:    30      ipReasmReqds:        0      ipReasmOKs:          0
ipReasmFails:      0      ipFragOKs:           0      ipFragFails:         0
ipFragCreates:     0
```

#### Die Link - Statistik ....

Bytes total	I	RR	REJ	RNR	%RR	%REJ	%RNR
Link to DB0GOE		06.03.94 06:38:27 -	06.03.94 21:03:18				
RX: 4373648	44196	23645	2154	188	30	2	0
TX: 14631652	78768	27066	238	220	61	0	0
Link to DB0AX		06.03.94 06:38:27 -	06.03.94 21:03:17				
RX: 1295328	16120	14472	636	0	42	1	0
TX: 6824718	33662	11907	6	0	73	0	0
Link to DB0NHM		06.03.94 06:38:27 -	06.03.94 21:03:15				
RX: 931191	5796	9626	663	1124	61	4	7
TX: 2915931	15564	6895	407	3	118	7	0

Bytes total    I    RR    REJ    RNR    %RR    %REJ    %RNR

Beschriftungszeile der Linkstatistikzahlen RX bzw. TX:

RX bzw. TX:

<anzahl> Bytes total	Byte insgesamt,
<anzahl> I	Infoframe,
<anzahl> RR	Bestätigungs-Frame,
<anzahl> REJ	Reject-Frame,
<anzahl> %RR	RR Frames/gesendete I Frames in %,
<anzahl> %REJ	REJ Frames/gesendete I Frames in %,
<anzahl> %RNR	RNR Frames/gesendete I Frames in %.

%RR:

Ziel ist es, nicht jedes einzelne I Frame zu bestätigen sondern nach Möglichkeit mehrere I-Frames mit einem RR-Frame zu bestätigen. Hierfür ist der T2-Timer zuständig. Andererseits darf aber auch die Gegenstelle nicht zu schnell pollen. Dieses Poll-Frame wird wiederum mit einem RR-Frame bestätigt.

%REJ:

Gibt Auskunft über die Qualität eines Links.

%RNR:

Gibt Auskunft über die Verarbeitungsgeschwindigkeit der Gegenstelle. RNR gibt es aber auch, wenn die Gegenstelle zu wenig Speicher hat.

%: Derzeit werden noch ALLE auf dem Interlink laufenden Frames ausgewertet. Ist der Level-2-Verkehr freigegeben, so werden natürlich auch REJ, RNR usw. von den Level-2-Verbindungen mit aufgenommen. Diesen Unterschied kann man aus der Statistik von DB0EAM besonders gut ablesen. Auf dem Interlink nach DB0AX ist im Gegensatz zu dem Interlink nach DB0GOE der Level-2-Verkehr nicht zugelassen. Nach DB0AX greifen die Steuerfunktionen des Level-3/4 Protokolles. Es kommt nur sehr selten zu einem RNR- Frame. Andererseits kann man hier auch den konsequenten HF-Technischen Aufbau (keinerlei gegenseitige Beeinflussung der Interlinks) von DB0EAM ablesen. Wurden von DB0AX in der Regel 2,4% der Pakete rejected, so sind es in der Gegenrichtung fast 0%. Auf dem Interlink nach DB0KH könnte das nur nach Sperrung des Level-2 festgestellt werden. Vergleicht man nun die Statistik von DB0NHM und DB0AX (sind zwar nur 23 Stunden, aber in der Tendenz gleichbleibend), so sieht man deutlich den gravierenden Nachteil eines Level2-Systems wie FlexNet. Die große Anzahl von RNR wird über ALLE Teilstrecken, über die der Level-2 aufgebaut wurde, übertragen. Auf dem Interlink nach DB0AX ist Level-2-Digipeating gesperrt, und damit steuert der Level-3/4 den Datenfluß. Genauso nachteilig verhält sich das Bestätigungsverhalten von FlexNet-Nachbar.

Link to <call> <datum> - <datum>.

Diese Rufzeichen-Statistik zeigt an, wann die Statistik zuletzt gelöscht worden ist, und wann das letzte Byte empfangen worden ist.

#### Die Port - Statistik ....

	Links	RxB	TxB	RxBaud	TxBaud	RxOver	TxOver
0:70cm_1200	6/5	5387k	52340k	64	208	83%	8%
1:70cm_9600	4/3	27434k	576083k	56	104	42%	1%
2:23cm_9600	4/1	10524k	60302k	392	248	6%	12%
3:Lohfelden	5/2	20441k	119742k	48	48	46%	9%
4:Cluster	23/1	68199k	31129k	128	48	10%	59%
5:Mailbox	21/1	2011M	588261k	4720	1616	2%	17%
6:Goettingen	1/1	176951k	424651k	496	1216	8%	2%
Total = 922.987.789 Bytes							

**In der Statistik werden nur die Info-Byte erfaßt!**

<Portnummer>:<Portname>

Port Nummer und Portname.

Links x/y

x = Anzahl der L2 Verbindungen

y = Anzahl unterschiedlicher Call.

Links wird für die Einstellung der automatischen Parameter gebraucht.

RxB ... TxB

Empfangene ... gesendete Datenmenge in Byte; k = Kilobyte; M = Megabyte; G = Gigabyte.

RxBaud ... TxBaud

Momentane Empfangsseitige ... Sendeseitige Linkbelastung.

RxOver ... TxOver

Prozentuales Verhältnis von Nutzdatenübertragung zu Protokolldatenübertragung.

Total = <anzahl> Bytes.

Summe aller empfangenen und gesendeten Byte.

(TA)LK <call>

Schaltet eine TALK-Verbindung zu dem User mit dem <call>, wenn <call> mit dem Kommandointerpreter verbunden ist. Als Antwort wird dieser Text ausgegeben: You are now talking with <call>. Leave this mode with /q. Alle weiteren Eingaben bekommt der User als: Msg from <call>: text..... Dieses ist jedoch nur eine einseitige Verbindung.

(TA)LK <call> <text>

Mit TALK <call> <text> kann man einem User eine Textzeile zusenden, wenn dieser mit dem Kommandointerpreter des Knotens verbunden ist. Besteht zu dem User keine Verbindung vom Kommandointerpreter, so wird die Textzeile bis zu einem Reconnect aufgehoben. Wird die Verbindung durch einen Disconnect vom User aufgelöst, wird die Textzeile .....fachgerecht und gebührenfrei entsorgt.

(TI)ME

Gibt das Systemdatum und die Uhrzeit aus.



(U)SER

Nach der Eingabe von U (für USER) erscheint z.B.:

KS:DB0EAM> TheNetNode (PC) V1.76	
Uplink(DK3NZ-5)	<--> Downlink(DK3NZ-10 DB0EAM-3)
Uplink Northeim via DB0NHM	
Circuit(LOH:DB0VFK DL2FAI-1)	<..> Downlink(DL2FAI-14 DB0EAM-3)
Uplink(DB0OVA-15)	<--> Downlink(DB0OVA DB0EAM-4)
Uplink Knuell via DB0KH DB0SHI	
Uplink(DL8FAJ)	<--> Downlink(DL8FAJ-15 DB0EAM-4)
Uplink(DD1SI)	<--> Downlink(DD1SI-15 DK5ZK)
Uplink Knuell via DB0KH DB0SHI DB0DQ	
Downlink 9600_Bit/s	
Convers(DB0EAM-1) Host	<--> Circuit(GOE7:DB0GOE DB0EAM-1)
Host(DB0EAM)	

KS Ist der „ALIAS“ des Knotens, eine Art Pseudonym. Zum einen soll dieser ALIAS eine geographische Information bieten, die einfacheres Zuordnen unbekannter Calls zu einem Gebiet ermöglicht. Zum anderen ermöglicht, im Gegensatz zum IDENT (Rufzeichen), der ALIAS ein mehrfaches Connecten eines Knotens, denn man kann z.B.:

KS, KS-1, KS-2,

etc. gleichzeitig connecten, aber nur einmal DB0EAM.

Die andere Möglichkeit, Multiconnect an einem TheNet-Knoten durchzuführen, ist die Benutzung des eigenen Rufzeichens mit verschiedenen SSID gleichzeitig. Das kann derzeit aber nicht jede Software. Als ALIAS wird in DL häufig des Autokennzeichens des Haupteinzugsbereiches verwendet, kurze Ortsnamen können auch vorteilhaft ausgeschrieben werden, wodurch der Erkennungswert weiter zunimmt. Andere Alternativen wie z.B. Postleitzahlen oder WW-Locatoren sind nicht sehr einprägsam.

Die Verwendung des ALIAS, statt des Rufzeichens, ist K E I N Rufzeichenmißbrauch, da der ALIAS ja gar keinen den internationalen Normen für Amateurfunkrufzeichen entsprechenden Aufbau hat (haben sollte...). Genausowenig wie ja auch ein Ruf an CQ oder Test kein Rufzeichenmißbrauch ist oder das 4-Zeichen Call bei Amtor. Die geforderte Rufzeichen-Nennung alle 10 Minuten wird vom Netzknoten mit einer Bakenaussendung an ID ausgeführt, in der auch der verwendete ALIAS mitgeteilt wird. Außerdem bleibt die Verwendung des ALIAS im AX25-Adreßfeld auf Endanwender begrenzt, zwischen den Knoten selbst werden die offiziellen IDENT (Rufzeichen) benutzt. Auch läßt sich jeder TheNet-Knoten anhand des INFO-, USER- und NODES-Befehles identifizieren. Eine einheitliche Verwendung von Autokennzeichen in Knotenlisten erhöht die Transparenz der Netze wesentlich.

DB0EAM

Ist das offizielle IDENT (Rufzeichen) des TheNet-Knotens.

Uplink <call>

Zeigt an, daß der Benutzer mit dem Rufzeichen <call> hier am Knoten in das Netz eingestiegen ist. Etwa vorhandene Digipeater werden in der nächsten Zeile angezeigt.

Downlink <call>

Zeigt an, daß der Benutzer hier mit dem Rufzeichen <call> das Netz verläßt. Etwa vorhandene Digipeater werden in der nächsten Zeile angezeigt.

Circuit <alias:ident>

Auf der linken Seite bedeutet, daß der Benutzer von dem Knoten mit dem <id> aus diesen Knoten connected hat und dort mit dem Rufzeichen <call> connected ist.

Circuit <alias:ident>

Auf der rechten Seite bedeutet, daß die Verbindung zu dem Knoten mit dem <id> geschaltet ist und der Benutzer dort das Rufzeichen <call> führt.

Host

Bedeutet eine Verbindung zum Bedienerterminal des Knotens oder auch zu einer direkt am Knoten angeschlossenen Mailbox.

<-->

Zeigt eine bestehende Verbindung an.

<..>

Zeigt eine im Aufbau befindliche Verbindung an.

Ein Eintrag ohne „rechte Seite“ bedeutet, daß die Verbindung hier zur Zeit endet und der Benutzer mit dem Befehlsinterpreter des Knotens verbunden ist.

**Uplink @ DB0FD**

Zeigt den Einstiegsknoten des Users an, sofern dieser nicht aus der Circuit-Zeile bereits hervorgeht. Dieses ist immer dann der Fall, wenn mehrere Level4-Verbindungen (Circuits) nacheinander aufgebaut werden.

**Downlink Port\_5 via DB0XYZ  oder  Uplink Port\_5 via DB0XYZ**

Zeigt den Downlink-Weg an, den der User genommen hat. >Port\_5< steht in diesem Fall für Port-Namen des entsprechenden Port (siehe **(PO)**rt Befehl) und ist ein Level2-Link. Ein Uplink- oder Downlink-Port wird nur angezeigt, wenn es sich dabei nicht um einen User-Port handelt, auf dem eine MHEARD-Liste geführt wird.

## (U)ser (H)ost

Wie (U)ser, zeigt jedoch nur die Verbindungen zum Host an. Wenn das Host - Interface zum Anschluß einer Box oder Cluster benutzt wird zusätzlich diese Liste ausgegeben.

CH	Call	F	NT	RX	TX	ST	RxB	TxB	Baud	ConTime
03	DB0IL-3	CD	7199	1	0	0	65	1494	120	0:02:05
05	DB0IL-5	C	7086	0	0	0	51	595	56	0:02:05
06	OZ5BBS-1	C	7068	0	0	0	7620	1598	96	0:09:37
/	/	/	/	/	/	/	/	/	/	/
1)	2)	3)	4)	5)	6)	7)	8)	9)	10)	11)

- 1) Hostmode-Kanal
- 2) Rufzeichen des Users (Connect vom User) oder eigenes Call (Connect vom Host zum Knoten)
- 3) Hostmode-Flags  
C = Connected  
D = Disconnecten wenn Info übertragen
- 4) Noactivity-Timer (in Sekunden)
- 5) empfangene Frames in Warteschlange
- 6) zu sendende Frames in Warteschlange
- 7) Statusmeldungen in Warteschlange
- 8) Anzahl empfangender Bytes seit Bestehen des Links
- 9) Anzahl gesendeten Bytes seit Bestehen des Links
- 10) Aus 8) + 9) errechnete effektive Baudrate für diesen Link
- 11) Connectzeit

## (U)ser +

Zeigt in Tabellenform die Level2-Verbindungen an.....

L2 - User:												
PO	SrcCall	DstCall	LS	Rx	Tx	Tr	SRTT	RxB	TxB	Baud	ConTime	Pri
1	DB0EAM	DB0GOE	IXF	0	9	0	79	109928k	685912k	1413	38:59:23	-
3	DG9ABF-15	DB0EAM-3	IXF	0	0	0	755	4583	9	253	0:02:25	-
3	DL1DAQ-13	DB0EAM-3	DBS	10	0	0	674	259713	188	200	2:52:29	-
4	DB0RDX-15	DB0EAM-4	DBS	8	0	0	172	50776	6325	15	8:06:15	-
5	DB0EAM	DB0AX	IXF	0	0	0	62	273953k	169538k	1330	38:59:17	-
6	DB0EAM	DG9XYZ	IXF	0	4	0	370	451216	189109	377	3:46:25	0
6	DB0EAM	DG9XYZ-1	IXF	0	0	0	96	104530	2955	33	7:11:00	20
/	/	/	/	/	/	/	/	/	/	/	/	/
1)	2)	3)	4)	5)	6)	7)	8)	9)	10)	11)	12)	13)

- 1) Benutzter Port zur einfacheren Übersicht.
- 2) Quellrufzeichen des L2-QSOs.
- 3) Zielrufzeichen des L2-QSOs.
- 4) L2-Link-Status:  
SET = Link-Setup,  
FMR = Frame Reject,  
DRQ = Disconnect Request,  
HTH = Wartet auf Connect eines Digipeating-Zieles,  
IXF = Info Transfer,  
REJ = REJ Send,  
WAK = Waiting Acknowledge,  
DBS = Device Busy,  
RBS = Remote Busy,  
BBS = Both Busy,  
WDB = Waiting Ack and Device Busy,  
WRB = Waiting Ack and Remote Busy,  
WBB = Waiting Ack and Both Busy,  
RDB = REJ Send and Device Busy,  
RRB = REJ Send and Remote Busy,  
RBB = REJ Send and Both Busy.
- 5) Anzahl der empfangenen Frames in der Warteschlange für diesen Link.
- 6) Anzahl der noch zu sendenden Frames in der Warteschlange für diesen Link  
Wenn ein Link zu einem Nachbarn zusammenbricht, wird dieser bei mehr als 100 ausstehenden Paketen im L2 einfach abgebrochen.
- 7) Anzahl Retries.
- 8) Stand des 'Smoothed Round Trip Timers'(in mal 10 ms).
- 9) Anzahl empfangender Bytes seit Bestehen des Links.
- 10) Anzahl gesendeter Bytes seit Bestehen des Links.
- 11) Aktuelle effektive Baudrate für diesen Link.
- 12) Connectzeit in HH:MM:SS und bei über 23:59:59 Stunden dann nur noch als TTT/HH:MM.

- 13) Bei DAMA-Netzeinstiegen:  
Aktuelle Priorität des User (0: = höchste Priorität).

.....und auch die Level4-Verbindungen.

L4 - User:											
Call	Node	S	Rx	Tx	Tr	Win	SRTT	RxB	TxB	Baud	ConTime
DL3EAG	DB0II	I	0	0	0	10	7698	0	12504	48	2:33:55
DB0EAM-1	DB0LIP	I.	0	0	0	10	914	68693	1400k	152	3/14:24
DB0KG	DB0GOE	.I	0	0	0	10	998	257	15781	456	0:02:56
DJ9KG	DB0GOE	I	0	0	0	10	7161	47	20079	48	2:39:36
/	/	/	/	/	/	/	/	/	/	/	/
1)	2)	3)	4)	5)	6)	7)	8)	9)	10)	11)	12)

- 1) Rufzeichen des User.
- 2) Call des Knotens mit dem der User verbunden ist.
- 3) L4-Circuit-Status.  
S = Circuit-Setup,  
.I = Eigener Knoten steht im Choke (Datenübertragung angehalten),  
I = Der Zielknoten steht im Choke,  
I = Normaler Info-Transfer,  
D = Disconnect-Request.
- 4) Anzahl der empfangenen Frames in der Warteschlange für diesen Circuit.
- 5) Anzahl der noch zu sendenden Frames in der Warteschlange für diesen Circuit.
- 6) Anzahl Transport-Retries.
- 7) Transport Fenstergröße.
- 8) L4 SRTT.
- 9) Anzahl empfangender Bytes seit Bestehen des Circuits.
- 10) Anzahl gesendeter Bytes seit Bestehen des Circuits.
- 11) Aktuelle effektive Baudrate für diesen Circuit.
- 12) Connectzeit in HH:MM:SS und bei über 23:59:59 Stunden dann nur noch als TTT/HH:MM.

Zur Messung des L4SRTT:

Durch den Aufbau des NET/ROM Layer 4 kann man Grundsätzlich nur in Senderichtung den SRTT messen, d.h. immer wenn wir Info senden, messen wir die Laufzeit. Von dem L4SRTT wird hauptsächlich das Acknowledge - Timeout und das Requery - Timeout abgeleitet.

Acknowledge - Timeout:

Der L4 kann Daten Senden, bis ein sogenanntes Sendefenster voll ist. Dieses umfaßt in der Regel 10 Frames. Das Sendefenster wird beim Verbindungsaufbau zwischen den Partnern vereinbart und ist auf beiden Seiten gleich. Danach wird auch das Acknowledge- (Bestätigungs-) verhalten ausgerichtet. Sobald das Empfangsfenster halb voll ist, wird sofort eine Bestätigung gesendet, spätestens aber nach der einfachen L4SRTT-Zeit. Damit wird einerseits ein ständiger Fluß sichergestellt, andererseits werden unnötige Bestätigungen vermieden.

Requery - Timeout:

"Requery" ist hier eigentlich nicht richtig, denn NET/ROM hat keine Kommandos in Empfangsrichtung. Dies bedeutet, das nach einer bestimmten Zeit das Frame vom Sender wiederholt werden muß, der Empfänger kann nicht mitteilen, das er das Frame nicht bekommen hat. Man spricht von Requery-by-Timeout, man könnte auch sagen Retransmission-Timeout. Dieser Timeout wird wieder vom L4SRTT abgeleitet, mit Faktor 3.

Was sagt uns das?

Ein sehr hoher SRTT-Wert (sagen wir 100s) bedeutet nicht, das diese Verbindung wirklich so lahm ist, sondern einfach das dort wenig los ist, und beide seiten sehr selten ein ACK schicken. Da wir sowieso beim halben Fenster ein ACK schicken, kann die Verbindung jederzeit schnellen Datentransfer aufnehmen. Andererseits ist es richtig, das solche Links mit sehr langen Requery - Timeout belegt werden, schließlich könnte es sein, das der Partner sich einfach lange Zeit läßt für die Bestätigung (hoher L4SRTT). Die hier vorgestellten Abläufe sind TCP/IP entliehen und den besonderen Anforderungen von NET/ROM angepaßt. Da TCP/IP mit Antwortzeiten im Millisekunden - Bereich arbeitet, wird sich erst zeigen, ob das alles so richtig hinhaut. Durch Begrenzungen nach unten und oben ist im Programm sichergestellt, das nicht allzuschiefe "automatische" Timerwerte generiert werden können.

**(U)ser (L)inks**

Zeigt in Tabellenform NUR die Level2-Verbindungen an.

**(U)ser (C)ircuit**

Zeigt in Tabellenform NUR die Level4- (Circuit -) Verbindungen an.

**(U)ser <Port Nr.>**

Zeigt in Tabellenform NUR die Level2-Verbindungen des angegebenen Ports an.

**(U)ser <call>**

Zeigt alle Informationen zu einem bestimmten User oder einer Gruppe von Usern an. Dabei sind die folgenden Abfragen möglich:

u db7\*;

u db7kg\* zeigt aber auch db7kga, db7kgb, db7kgc usw.

aber auch alle SSID von DB7KG;

u db7kg zeigt db7kg-0 an;

u db7kg-15 zeigt db7kg-15an;

u db7kg-\* geht NICHT

```

L2 - User:
Po SrcCall   DstCall   LS  Rx Tx Tr SRTT   RxB   TxB   Baud   ConTime Pri
-----
 2 KS-5      DG9FU-5   IXF  0  0  0   95  2624  96563   96   2:13:33  -
 2 KS-6      DG9FU     IXF  0  0  0   85   176   5367   72   1:07:41  -

Uplink(DG9FU-5)                <-->  Convers(DG9FU-5) Ch 170
Uplink 23cm_9600
Uplink(DG9FU)
Uplink 23cm_9600
DG9FU de DB0EAM (16:52) >

```

**(V)ERSION**

Gibt die aktuelle Versionsnummer der Software aus.

**(V)ERSION +**

Zeigt die Version und die Compiler Switches an.

Zur Zeit sind möglich:

DIPROUTE	IP-Router eingebunden
DPACSAT	PacSat Server eingebunden
DSYSPORT	Port Mode „Sysop“ ist freigegeben.

**EXTERNE Befehle für alle User**

Im Prinzip können nun die Filenamen frei gewählt werden. Textnamen und Programmnamen müssen sich jedoch MEHR als durch die Extension unterscheiden. Ein Programm MSG.EXE und ein Textfile MSG.TXT sind nicht möglich, es wird in diesem Fall nur das EXE-File gefunden.

Unter externe Befehle fallen nun alle Texte, die sich im PATH TEXTCMD befinden. Durch das Editieren der CTEXT.\*\*\* können den Usern nun weitere Texte angeboten werden. Zur Vereinheitlichung sollten bitte die folgenden Bezeichnungen beibehalten werden:

**(A)KTUELL**     (externer Befehl)

Gibt aktuelle Informationen aus.

**(HA)RDWARE**   (externer Befehl)

Gibt eine Hardwarebeschreibung des Knotens aus.

**(I)NFO**        (externer Befehl)

Ausgabe eines Info-Textes.

**(MAP)**         (externer Befehl)

Zeigt eine kleine Karte der direkten Umgebung des Knotens.

**(SO)FTWARE**   (externer Befehl)

Gibt eine Softwarebeschreibung aus.

**EXTERNE Programme für alle User**

Hier stehen im Moment die Programme:

**Digimail:**

(MSG) (externer Befehl)

(MSG) S <ZIELCALL/GRUPPE> #<lifetime> <text>  
Zum Erstellen einer Digimail:

Der MSG-Befehl ermöglicht es, einem Benutzer des Knotens eine kurze (!) Nachricht in den Connect-Text zu schreiben. Beim nächsten Connect wird diese Zeile dann bei ihm im CTEXT erscheinen. Statt des Zielcalls kann auch eine Zielgruppe angegeben werden. Diese ist eine Art Verteilerliste. Jedes Mitglied der Zielgruppe bekommt eine Kopie der Nachricht.

Die Lifetime kann, muß aber nicht, angegeben werden. Sie kann von #1 = einem Tag bis #99 = neunundneunzig Tagen angegeben werden. Wird keine Lifetime angegeben, so wird sie auf default = 14 Tage gesetzt. Das Herabsetzen der Lifetime geschieht mittels MSY C. Siehe „Externe Programme für den SYSOP“.

Abhängig davon, ob das System mit einer Festplatte oder einer RAMDISK arbeitet, kann die Nachricht bei Absturz oder Reset verloren gehen.

Beispiel: MSG S DL9GYA #10 Roland, bitte connecte mich, wenn Du zurück bist!

MSG S SYSOP Link nach ..... defekt?!

Letzteres würde die Nachricht an DL1KWS und DL9GYA weiterleiten. Die Definition einer neuen Gruppe geschieht nur durch den Sysop. Bitte danach fragen.

(MSG) R  
Liest die eigenen Nachrichten aus.

(MSG) R <call>  
Liest die an <call> gerichteten Nachrichten aus.

(MSG) R <call> 1 oder 1-3  
Zusätzlich kann noch eine Numerierung mit angegeben werden. Also Lesen der Nachrichten 1 oder 1-3.

(MSG) L  
Listet die eigenen Header auf.

(MSG) E  
Der MSG E Befehl löscht alle eigenen Digimail-Nachrichten im eigenen CTEXT.

(MSG) G  
Zeigt alle vorhandenen Verteilergruppen mit Call an.

(MSG) G <gruppe>  
Zeigt nur die Call in dieser <gruppe> an.

Neue Gruppen können nur vom Sysop angelegt werden!

(MSG) V  
Gibt die Versionsnummer und - Datum aus.

**Locatorberechnung:**

(QTH) (externer Befehl)

**(QTH)** <locator> oder **(QTH)** <locator> <locator>

QTH berechnet Entfernung und Richtung zwischen zwei QTH-Kennern. Wird nur ein QTH-Kenner angegeben, so wird die Entfernung zum Knoten berechnet ! Außerdem ermittelt QTH alle gültigen Angaben für die Standorte, z.B. zur Ermittlung des neuen weltweiten QTH-Kenner aus der geografischen Koordinate. QTH akzeptiert sowohl den neuen weltweiten als auch den alten QTH-Kenner. Außerdem können Längen- und Breitengrade in Dezimalform oder auch in Grad:Minuten:Sekunden angegeben werden.

**(QTH)** <locator>

Berechnung der Entfernung und Richtung zwischen dem Standort des Knotens und <QTH>.

**(QTH)** <locator1> <locator2>

Berechnung der Entfernung und Richtung zwischen <locator1> und <locator>.

Gültige Eingabeformate:

Alter QTH-Kenner : FM04C oder FM04C/2 (Ohne Angabe gilt Feldraster /2)

Neuer QTH-Kenner : JO52JW

Geogr. Koordinate: GGG:MM:SS/GG:MM:SS z.B. 10:47:30/52:56:15

(östl.Länge/nördl.Breite in Grad:Min:Sek)

Geogr. Koordinate: GGG.GGG/GG.GGG z.B. 10.792/52.937

(östl.Länge/nördl.Breite in Dezimalform)

Es ist zu beachten, daß innerhalb einer geografischen Koordinate nur in Grad:Minuten:Sekunden oder mit Gleitkommazahlen (Realzahlen) gearbeitet werden kann. Ein Mischen beider Formate ist unzulässig.

### **SATELLITEN-Standort-Berechnung:**

**(SAT)** (externer Befehl)

Zeigt die derzeit 40 möglichen Satelliten sowie eine kurze Hilfe an.

**(SAT)** <nr>

Berechnet die Satellitenposition zum Zeitpunkt der Abfrage, bezogen auf den Standort des Knotens. Ausgegeben werden nicht nur die Positionsdaten, sondern auch die nötigen Antennen-Einstellungen.

**(SAT)** <nr> <locator> oder **(SAT)** <nr> <koordinaten>

Wie zuvor, jedoch wird nicht der Locator des Knotens, sondern der angegebene verwendet.

Gültiges Eingabeformat:

Geogr. Koordinate: GGG.GGG GG.GGG z.B. 10.792 52.937

(östl.Länge/nördl.Breite in Dezimalform)

### **User-Statistik:**

**(TOP)** (externer Befehl)

Dieser Aufruf wertet die MHEARD.TAB aus und erzeugt eine „Hitliste“. Die Auswertung ist nicht immer auf dem neuesten Stand. Die Datei MHEARD.TAB wird in Abhängigkeit des Parameters 11 aktualisiert. Der aktuelle Stand kann an der Uhrzeit im Kopf der Tabelle abgelesen werden.

**(TOP)** <anzahl>

Gibt eine Liste mit der Länge bis maximal <anzahl> aus.

**(TOP)** -<port>

Gibt eine Liste von 10 Calls, die auf dem <port> bekannt sind, aus.

**(TOP)** <anzahl> -<port>

Gibt eine Liste mit der Länge bis maximal <anzahl>, die auf diesem <port> bekannt sind, aus.

**(TOP)** <call>

Gibt nur dieses Rufzeichen, aber mit den verschiedenen registrierten SSID aus.

**(TOP)** <ca\*>

Z.B.: TOP DG8B\* . Gibt alle Rufzeichen aus, die mit DG8B anfangen.

**(TOP)** -L3

Wertet statt der MHEARD.TAB die L3HEARD.TAB aus. In der werden die Links gespeichert. Der Parameter -l3 kann mit -port und anzahl verknüpft werden.



**(TOP) -h**

Gibt eine kurze Hilfe aus.

**(TOP) -v**

Zeigt den Versionsstand an.

MHEARD-Auswertung vom 01.11.97 08:33 UTC bis 07.11.97 22:07 UTC										
Nr.	Call	Port	Rx	Tx	Summe	%	RxRej	TxRej	Dama	Son.
1	DB0CXH	CUX	5700751	40803866	46504617	47	571	6931	0	*
2	DB0WHV	WHV	14771265	8463332	23234597	23	865	881	0	
3	DL3BCO	70cm	6702818	157864	6860682	6	32	0	4	
4	DB0DIM	CUX	1479456	3923212	5402668	5	62	2919	0	#*
5	DG3BAF	70cm	2757348	175502	2932850	2	12	0	0	
6	DG8BR	70cm	1969490	269554	2239044	2	64	0	0	#
7	DG9BJY	70cm	2014585	48985	2063570	2	1	0	0	

```
. Hat Call empfangen ---! ! ! ! ! ! !
. Hat Call gesendet -----! ! ! ! ! ! !
. Hat Call über den Digi übertragen -----! ! ! ! ! ! !
. Hat Call Anteil an Digiauslastung -----! ! ! ! ! ! !
. Hat Call gesendet, weil schlecht gehört -----! ! ! ! ! ! !
. Hat Call empfangen, weil Digi schlecht gehört -----! ! ! ! ! ! !
. Hat Call gegen DAMA verstoßen -----! ! ! ! ! ! !
. Call hat * = VIA gearbeitet -----! ! ! ! ! ! !
. Call ist # = mit unterschiedlichen SSID auf Ports QRV -----! ! ! ! ! ! !
```

103	DL6BI	70cm	1333	15	1348	0	0	0	0	
104	DG6BAF	70cm	975	135	1110	0	0	0	0	
105	DK5BO	70cm	773	190	963	0	0	0	0	
106	DB0MHD	CUX	87	355	442	0	0	0	0	

Ausgewertet wurden alle Ports.

Die aufgelisteten Benutzer haben 98.405.235 Bytes = 100 % Übertragen.

Insgesamt wurden 98.405.235 Bytes von 106 Usern bewegt.

Falls sich jemand über die Differenz zwischen Plätzen und Usern wundert: Sie entsteht wenn ein Benutzer auf mehreren Ports aktiv ist. Z.B.: 70cm und 23cm. Oder so... Tatsache ist, es wird das Rufzeichen nur einmal gezählt.

Beispiel für eine Einzelausgabe:

Datum	Port	Rx	Tx	RxRej	TxRej	D	Call	Via
31.03.95	22:53 70cm	7528028	1917008	120	0	0	DG3BAF	
31.03.95	22:32 70cm	314818	3684301	13	0	0	DG3BAF-2	
30.03.95	10:08 70cm	0	5	0	0	0	DG3BAF-15	DL1BKL
25.03.95	09:49 70cm	23946	65179	1	0	0	DG3BAF-1	
03.03.95	10:07 70cm	1175	438	0	0	0	DG3BAF-4	

Die Liste erklärt sich wohl von selbst.

Unter D stehen die DAMA-Verstöße, und unter Via steht der, der sich als Digipeater zur Verfügung gestellt hat.

**Visitenkarte:****(SAV)ecall** (externer Befehl)**(SAV)ecall** <feldkennung> <text>Die Felder können mit **(SAV)ecall** ausgefüllt oder geändert werden.

SAVECALL	/N	.....	30	Zeichen für den Namen
SAVECALL	/Q	.....	30	Zeichen des Wohnortes
SAVECALL	/L	.....	6	Zeichen des World-Locators
SAVECALL	/D	.....	7	Zeichen für Eingabe des DOK
SAVECALL	/V	.....	9	Zeichen QRV auf ... Digi
SAVECALL	/M	.....	25	Zeichen Mybbs der Mailbox
SAVECALL	/T	.....	40	Zeichen für freien Text
		( ein * als Text löscht diesen Eintrag )		
SAVECALL	/*	==>		Vorsicht !! löscht den ganzen Eintrag !

DOK Bitte so eintragen: F73 oder F73/Z25

.....sonst klappt es mit dem Suchen nachher nicht.

(SH)owcall (externer Befehl)

(SH)owcall <call>

Ist ein kleines Programm, welches die Visitenkarten der User, die sie sich selbst SAVECALL erzeugt haben, ausliest und anzeigt. Muster:

```
== ShowCall ===== 01.09.95 ==
= Call   : ..... Name : ..... =
= Loc    : ..... QTH  : ..... =
= QRV on : ..... MyBBS : ..... =
= Dok    : ..... Update : JJMMTTHH CALL =
=                                     =
= Free MSG : ..... =
===== de DG3AAH =====
```

Die Liste lebt von der Beteiligung der User. Deshalb darf nur der darin lesen und suchen der selbst zumindest die Felder Name, Locator und QTH eingetragen hat. Das <call> kann auch in Verbindung den Platzhaltern „\*“ und „?“ aufgerufen werden. Dabei steht „\*“ für beliebig viele Zeichen und „?“ steht für genau 1 Zeichen. Es wird eine Liste der Calls aus, bestehend aus :

- Call, Name, Wohnort und Locator.

Die ersten 3 Zeichen müssen jedoch angegeben werden, damit nicht unendliche Listen ausgelesen werden.

(SH)owcall W

Zeigt die Anzahl der Einträge an.

### **Onlinehilfe:**

(H)elp (externer Befehl)

Ersetzt die alte Hilfsfunktion 'H'. Es wird nur noch in einer Datei nach dem eingegebenen Befehl gesucht. Zeigt alle möglichen Befehle an.

(H)elp <befehl>

Zeigt die erste Seite (20 Zeilen) der Hilfe zum <befehl> an.

(H)elp <befehl> 2

Zeigt die zweite Seite der Hilfe zum <befehl> an.

(H)elp <befehl> \*

Zeigt alle Seiten der Hilfe zum <befehl> an.

## Framesammler

TNN kennt derzeit nur die AX.25 Level 2 Protokollversion 2.0. In dieser Version ist kein Framesammler spezifiziert. Statt dessen ist es zulässig, nicht bestätigte Frames zu verändern. Also wenn ich die Frames 0 bis 6 sende, Du aber nur Frame 0 und 6 empfangst, darf ich anschließend den Inhalt der Frames 1 bis 6 ändern. Das macht TF zwar nicht, aber TF ist nicht das Maß aller Dinge, sondern eher schon AX.25. Wenn man (AX.25) solche Änderungen des Frameinhalts zuläßt, ist aber ein Framesammler gleichzeitig ausgeschlossen. In obigem Beispiel hättest Du Dir ja den Inhalt von Frame 6 gemerkt, aber eben diesen Inhalt ändere ich nun. Das Ergebnis ist verständlicherweise MURKS. Anders ist die Situation, wenn TNN einmal umgestellt wird auf die AX.25 Level 2 Version 2.2. In dieser neuen Protokollversion ist neben anderen Neuerungen auch ein selektives Rejecten von Frames spezifiziert, was einer Framesammlerfunktion entspricht. Dieses darf aber nur verwendet werden, wenn über eine entsprechende Prozedur eine Parametervereinbarung zwischen den Stationen durchgeführt wird. Falls eine der Stationen dieses selektive Rejecten nicht unterstützt, wird es von keiner der beiden Stationen verwendet, und außer der Reihe empfangene Frames sind zu verwerfen. Andererseits muß man sich einmal überlegen, weshalb der Framesammler überhaupt zum Einsatz kommen sollte. Wenn ein User ein Frame nicht mitbekommt, gibt es dafür einen Grund - und da kann ich mir gleich mehrere vorstellen.

- a) Jemand anders hat einfach dazwischengesendet. Das sollte wohl nur sehr selten vorkommen, denn kein PR-Aufbau darf auf Sendung gehen, wenn der Digi oder ein anderer User sendet. Das trifft dann zwar einen Unschuldigen, für den evtl. mehr wiederholt werden muß, als nach Framesammlerphilosophie notwendig, aber trotzdem sehe ich hier keinen Handlungsbedarf.
- b) Die Empfangsanlage des Users ist fehlerhaft - also entweder der Rx ist neben der Digipeaterfrequenz, oder das Modem hat eine Macke. Hier trifft es genau den richtigen - wenn es trotzdem schön schnell geht, gibt es keinen Grund zur Verbesserung der Empfangsanlage .. also eher gegenteilige Handlungsweise: wenn vom Digi ein REJ empfangen wird, Durchsatz drastisch zu Gunsten anderer User verringern. Wegen Fall a) kann das natürlich ziemlich ungerecht sein, aber (s.o.) das tritt wohl nur selten auf. Also besser kein Framesammler!
- c) Die Empfangsverhältnisse beim User sind mies (schwankende Feldstärke). Das ist natürlich ähnlich, wie bei b), wenn auch hier evtl. schlecht versorgte Gebiete eine gute Ausrede darstellen. trotz allem sollte man dann mit einer besseren Antenne die Empfangsverhältnisse verbessern (oder durch eine Digipeater Spende in Form einer PA). Schließlich machen wir immer noch Amateur - FUNK, und dabei geht es ja hauptsächlich darum, mit dem Medium Funk zu experimentieren - mit Software kann man die Empfangsverhältnisse jedenfalls nicht verbessern. Also auch hier eher Durchsatz zu Gunsten von anderen Usern verringern, wenn ein REJ empfangen wurde.
- d) Der Digipeater-Sender ist fehlerhaft (Modem defekt oder Tx neben der Soll QRG). Wie bei b) sollte natürlich der Fehler behoben werden. Wenn es trotzdem noch flott geht, merkt keiner den Fehler, also besser kein Framesammler.

Soweit zum Framesammler beim User (also TF) und einer dazu passende Reaktion des Digipeater (nur angeforderte Frames wiederholen). Beim Digi ist das aber ähnlich. Auch hier gibt es verschiedene Möglichkeiten, warum ein Frame eines Users nicht empfangen wird.

- e) Jemand anders sendet dazwischen wie schon unter a). Dies passiert wahrscheinlich häufiger einmal, nämlich immer dann, wenn ein neuer User connecten will, der den gerade sendenden User nicht empfängt. Da ist eigentlich niemand schuld, außer bei DAMA vielleicht das DAMA-Protokoll selbst, das nicht einen Zeitschlitz vorsieht für neu einsteigende Stationen. Hier sollte man eher das DAMA-Protokoll modifizieren, als einen Framesammler einbauen, weil damit Kollisionen grundsätzlich auch beim Connecten vermieden werden. Schließlich können ja auch andere Frames als I-Frames zerstört werden.
- f) Der Digipeater-Empfänger ist fehlerhaft (falsche Frequenz / Modem defekt). Auch hier ist Reparatur angesagt, nicht ein Workaround per Framesammler. Das trifft zwar alle, aber eben nur bis zur Reparatur. Und je schlechter es geht, desto eher merkt jemand, daß der Digi kaputt ist.
- g) Der User Tx ist fehlerhaft (falsche Frequenz / Modem defekt). Wie bei b) sollte der Durchsatz drastisch reduziert werden, wenn ein REJ vom Digipeater gesendet wird - nur so merkt der User, daß seine Anlage nicht ok ist. Also wieder kein Grund für einen Framesammler.
- h) Der User wird beim Digipeater schlecht empfangen. Wie bei c) ist der User dafür zuständig, die Empfangsfeldstärke am Digi zu erhöhen (mehr Antenne zusätzliche PA). Auch hier besser kein Framesammler!

Letztlich bleibt also nur Fall e) als Begründung für einen Framesammler. Das ist aber meines Erachtens nicht ausreichend. Daß es nicht schön ist, wenn Frames unnötig wiederholt werden, ist schon klar. Ich finde es aber trotz allem nicht sinnvoll, so etwas problematisches wie einen Framesammler bei der AX.25 Version 2.0 zu implementieren. Selbst bei AX.25 Version 2.2 sehe ich es nicht als sinnvoll an, das selektive Rejecten zu bei einem Digipeater zu unterstützen (Begründung s.o.), aber es gehört bei der neuen Version nunmal dazu. Zwischen Einzelstationen ist das natürlich ganz etwas anderes. Für TF sollte das also auf jeden Fall vorgesehen werden - aber bitte nicht mit AX.25 Version 2.0.

**ANHANG A:****Die verschiedenen Textdateien und ihre Bedeutung**

Um die Gestaltung der Info - oder Hilfstexte möglichst einfach zu handhaben, haben wir den Weg gewählt, diese Texte als einzelne Dateien auf Diskette, RAM-Disk oder Hard-Disk abzulegen. Jeder einzelne Text kann vom SYSOP mit dem EDIT-Befehl editiert und mit dem READ-Befehl gelesen werden. Außerdem sind fast alle Texte auch anderen Benutzern über die einzelnen Kommandos zugänglich.

Folgende Texte sollten vorhanden sein:

**AKTUELL.TXT (ins SUB-DIR: TEXTCMD)**

Aktuelle Informationen zum Knoten, Veranstaltungen, usw.  
(Alte AKTUELL.TXT dürfen ruhig auch mal entfernt werden.)

**CTEXT.TXT (ins SUB-DIR: TEXTHLP)**

Connect-Text (wird nur ausgesendet, wenn freigegeben).  
Im CTEXT.TXT kann der Hinweis auf einen vorhandenen AKTUELL.TXT vorhanden sein. Soll dessen Datum automatisch mit angegeben werden, so ist dieses mit einem %A auszugeben. Beispiel für einen CTEXT:  
(H)elp (A)ktuell (vom %A)

**CTEXT.n (ins SUB-DIR: TEXTHLP)**

Portabhängiger Ctext. CTEXT.n wird zusätzlich zum CTEXT.TXT auf dem jeweiligen Port gesendet.

**HARDWARE.TXT (ins SUB-DIR: TEXTCMD)**

Informationen zum Knotenrechner, usw. (HARDWARE-Befehl).

**INFO.TXT (ins SUB-DIR: TEXTCMD)**

Informationen über Linkstrecken, usw. (INFO-Befehl).

**MAP.TXT (ins SUB-DIR: TEXTCMD)**

Kleine Karte der Nachbarknoten (MAP-Befehl).

**POLLDISC.TXT (ins SUB-DIR: TEXTHLP)**

Meldung, die gesendet wird, wenn der User weder DAMA noch Pseudo-DAMA macht. Wird nach Poll-UI-Count (PARAMETER ..) UI-MSG gesendet. Ein im Text vorhandenes '%n' wird in die gemessene Frackzeit umgewandelt.

**QUIT.TXT (ins SUB-DIR: TEXTHLP)**

Wird beim QUIT-Befehl ausgegeben, falls vorhanden.

**SOFTWARE.TXT (ins SUB-DIR: TEXTCMD)**

Informationen zur Software (SOFTWARE-Befehl).

**SUSPEND.TXT (ins SUB-DIR: TEXTHLP)**

Siehe dazu beim Suspend Befehl!

Eine Übersicht über alle vorhandenen Texte bekommt man eventuell mit dem 'HELP INDEX-Befehl. Dieser Befehl listet alle vorhandenen Texte in der Reihenfolge auf, in der sie auf Diskette stehen. Falls die (externe) Onlinehilfe installiert ist, steht der Befehl „HELP INDEX“ nicht zur Verfügung.

**ANHANG B:****(TNC2c)****TNC2c und TheNetNode**

Taktrate des SIO-SyncB-Taktes

Wenn der TNC mit 2.4576 MHz betrieben wird, dann unbedingt die Brücke BR1 durchkratzen und SyncB (SIO Pin 29) an Pin 1 des 74HC4060 legen. Diese Änderung ist auch in der TNC2c-Anleitung beschrieben. Soll der TNC mit 4.9152 MHz betrieben werden, dann sind folgende Änderungen erforderlich:

Quarz tauschen (... klar)

Brücke BR1 durchkratzen und SyncB (SIO Pin 29) - das ist der mittlere Anschluß von BR1 - an Pin 2 (Pin 2!!) des HC4060 legen. Der Pin 2 ist nicht an BR1 herangeführt, aber die Änderung ist notwendig, damit der SyncB-Takt 600 Hz beträgt.

Das EPROM muß eine Zugriffszeit von höchstens 200ns haben!

Soll der TNC mit 9600Bd auf der HDLC-Seite betrieben werden, so ist mindestens eine Taktfrequenz von 4.9152 MHz notwendig, besser sind 9.8304 MHz (Bei Vollduplex sind 9.8304 MHz zwingend erforderlich). Für den Umbau auf 9.8304 MHz werden ein paar Bauteile benötigt, die nicht jeder in der Bastelkiste liegen hat. Für einen Anwender-TNC mögen ausgesuchte Standardbausteine ausreichen, die jenseits ihrer Spezifikation betrieben werden. Der Netzknoten soll jedoch im Dauerbetrieb laufen, meist an einem nicht beliebig schnell erreichbaren Standort. Deshalb sollte man kein Risiko eingehen und die Bauteile nur innerhalb ihrer Spezifikation betreiben. Welche Art von Programmabstürzen oder auch nur Programmungenauigkeiten bei nicht spezifiziertem Betrieb auftreten, ist nicht vorherzusehen. So war z.B. bei einem TheNet-Netzknoten die Laufzeitermittlung falsch.

Für die CPU und die SIO sind 10 MHz-Bausteine zu verwenden (z.B. Zilog Z84C0010 und Z84C4010).

Das RAM benötigt eine Zugriffszeit von 80ns oder weniger.

Das EPROM benötigt eine Zugriffszeit von 100ns oder weniger.

Die Bausteine 74HC132 und 74HC139 sind unbedingt gegen 74AC132 und 74AC139 zu tauschen!

Vom 74AC139 (5) kann, wenn der TNC ohne AKKU betrieben wird, eine direkte Verbindung zum 62256 (20) hergestellt werden. Dadurch entfällt die Laufzeit über die beiden Gatter des 74HC132.

Die Brücke BR1 wird so verbunden, daß der SyncB-Anschluß der SIO (Pin 29) mit dem 74HC4060 Pin 3 verbunden ist.

KISS-Software für den TNC im Tokenring und Watchdog :

Aktuell ist das auf der Diskette befindliche KISS17.BIN.  
Es enthält den Eprominhalt für die TNC.

Adr. Zelleninhalt	Beschreibung (Alle Werte in HEX):
<b>5C 00..08..0F</b>	<b>Ringadresse je nach Prg. 8 ... 16 Kanäle,</b>
5D 00..FF	TXDelay in 10 ms Schritten HEX 14 = 200 ms,
5E 00..FF	PERSIST,
5F 00..FF	SLOTTIME,
60 00..FF	TXTAIL,
61 00..01	00 = Simplex; 01 = Vollduplex,
62 00..FF	PTT Haltezeit (Low-Byte),
63 00..FF	PTT Haltezeit (High-Byte),
<b>64 00..01</b>	<b>00 = DAMA aus; 01 = DAMA ein.</b>

Die Adressen 5C und 64 **müssen** beim Brennen des Eprom angepaßt werden. Die Adresse 62 und 63 ist bei einem VOLLDUPLEX-Link relevant. Die anderen Adressen werden von TNN aus eingestellt und dienen hier nur der vollständigen Information.

Adr. 5C: Jeder TNC im Ring muß seine eigene Adresse haben, die im EPROM stehen muß.

Adr. 5D..61: Diese Speicherzellen werden im RAM von TNN aus nach einem Neustart des TNC gemäß der eingestellten Parameter geändert.

Adr. 62 + 63: Die PTT Haltezeit ist nur bei der Einstellung VOLLDUPLEX aktiv. Ist der TX inaktiv, so wird er mit dem eingestellten TX-DELAY getastet. Nach Ende der Tastung läuft nun dieser Timer ab. Ist ein weiteres Paket in dieser Zeit zu senden, so wird es mit TX-DELAY = 0 ms ausgesendet und der Timer neu gesetzt. Hierdurch wird viel Sendezeit eingespart, die nun der Datenübertragung zur Verfügung steht. Soll diese Funktion ausgenutzt werden, so muß noch der 22 µF Kondensator am PIN 11 (SIO RTS NOT) des Modem-Disconnect auf der TNC2c Platte überbrückt werden. Die „Wachhund-Schaltung“ für die Sendertastung ist dann natürlich außer Betrieb.

Zu beachten ist, daß der TNC2 immer auf 32 KByte RAM aufgerüstet sein muß, und es sollten für 9600 Baud-Vollduplex-Stecken nur solche TNC eingesetzt werden, die mit einem CPU- Takt von > 6 MHz arbeiten!

Siehe hierzu eine Umbauanleitung unter ANHANG B.

Beispiel:Adr. 62 = Hex 50 ; Adr. 63 = Hex 46

Gesamt:Hex 4650 = 18000 ms = 3 Minuten Haltezeit.

Adr. 64: Ist das DAMA-Bit gesetzt, so teilt der TNC dem Rechner mit, wann das Paket gesendet wurde. Hiermit wird der DAMA-Tout (22) Timer gestartet. Es ist also zur Entlastung des Tokenringes und des Rechners nur bei dem TNC zu programmieren, der auch als DAMA-TNC eingesetzt werden soll.

**Watchdog, Version mit MAX 691:**

Solange der TNC arbeitet, gibt er an dem Punkt 5 der RS232 einen Wechsel zwischen + 10 Volt und - 10 Volt ab. Dieser Wechsel wird einem MAX 691 zugeleitet, der ihn überwacht. Bleibt der Wechsel aus, so wird der Ring durch ein Relais erst mal wieder geschlossen. Über den Punkt 8 der RS232, der mit dem Reset-Transistor des TNC zu verbinden ist, wird nun der TNC durch den MAX 691 neu gestartet. Kommen nun wieder Impulse vom Punkt 5 der RS232, so wird der TNC wieder durch Öffnen der Relaiskontakte in den Ring aufgenommen. Um den Max 232 mit zu überwachen, habe ich erst hier den Pegel durch eine Widerstands-Dioden-Zenerdioden-Widerstands-Kombination auf einen Pegel von 0 Volt und + 5 Volt umgesetzt, da der MAX 691 leider nur +5 Volt verträgt.

**Watchdog, Version mit 4060:**

Es werden die Datenleitungen zu den TNC und zum Rechner überwacht. Dies geschieht mit zwei preiswerten Bausteinen 4060. Der Baustein ist eigentlich ein Teiler. Er wird so beschaltet, daß er ständig zählt. Erreicht der Zähler einen Stand, so daß Pin 3 (Q14) aktiv wird, so wird ein Relais angesteuert, das entweder den Rechner resetet oder den TNC die Spannung entzieht. Der Teiler zählt weiter, und wenn Pin 13 (Q9) aktiv wird, dann bekommt der 4060 einen Reset-Impuls, der an Pin 12 geführt wird. An diesen Pin wird auch das Signal der RS232 geführt, was dazu führt, daß der 4060 ständig resetet wird, wenn Daten kommen. Pro Sekunde laufen in der Regel 20-40 Token über den Ring. Das dürfte reichen.

**Watchdog, Version mit VANESSA:**

Die Vanessakarte bietet mit dem PIN X38 auch eine RESET-Leitung an. Dieser Reset wird immer dann aktiviert, wenn 10 Minuten lang von der Karte (beide Ports) KEIN Frame gesendet wurde. Ist die Karte auf einem Interlink installiert, so wird dort ja alle 3 Minuten eine Bake gesendet und der Timer somit zurückgesetzt.

Der wesentliche Vorteil liegt darin, daß der Watchdog wesentlich günstiger (ca. 10.-DM) ist als die alte Version, die pro TNC/Rechner einen Aufwand von ca. 25.-DM erfordert. Der alte Watchdog soll manchmal nicht reagiert haben, da das Toggeln des RTS-Pins vorhanden war, obwohl sich der TNC aufgehängt hat.

Als Nachteil ist zu bewerten, daß man zum Knoten fahren muß, wenn tatsächlich mal ein TNC ausfällt, da dann der ganze Knoten steht und nicht nur ein Link betroffen ist. Aber nach Murphy trifft es eh immer den Link, der niemals ausfallen darf. Das ganze ist wahrscheinlich eine Glaubenssache, aber das Geld, das man spart, kann man besser in einen Link stecken.

**Wenn man noch einen „Not-Reset“ für den gesamten Netzknoten z.B. per DTMF-Steuerung vorsieht, kann man auch einen gezielten Reset durchführen und so einige Ausflüge zum Knoten sparen.**

**Und nach dem Neustart...**

Nach dem Neustart des TNC, hängt er an das nächste Token seine Neustartmeldung an. Daraufhin bekommt er vom Rechner seine ihm zugedachten Parameter wie Halb- oder Vollduplex, TXDelay usw. mitgeteilt. Gleichzeitig wird der Vorfall in der Statistik des Programmes aufgenommen, um solchen Störenfrieden auf die Schliche zu kommen.

**ANHANG C:****V A N E S S A****Installationsanleitung:****Inhalt:**

1. Allgemeines.
2. Spezifikation.
3. Installation und Konfiguration.
4. Probleme und Fehlersuche.

**1. Allgemeines:**

SEPRAN ist ein Akronym und steht für

SWISS EXPERIMENTAL PACKET RADIO AMATEUR NETWORK.

Unter dieser Projektbezeichnung fallen verschiedene Aktivitäten, die im Rahmen der SWISS-ARTG (Swiss Amateur Radio Teleprinter Group) ausgeführt werden. Die Arbeiten umfassen Tätigkeiten auf verschiedenen Ebenen wie Computer, Software, Antennen oder Hochfrequenz-Baugruppen um ein schnelles PR System verwirklichen zu können.

**1.1 Konzept PR-Knoten.**

Es gibt viele Ideen, wie eine leistungsfähige Hardware aussehen müßte. Insbesondere kann man sich auch an am Markt etablierten X.25-Produkten (Multi-PAD, Multiplexer) orientieren. Die Merkmale eines guten Vermittlungssystems sind sowohl die hohe Datenrate als auch der große Datendurchsatz. Der kritische Punkt des Systems wird immer erreicht, wenn sehr viele und kurze Datenpakete zu verarbeiten sind. Eine Aufteilung der Aufgaben in Routing einerseits und Unterhalt der Links andererseits wird als notwendig erachtet, um die Ziele zu erreichen.

Ideal erscheint ein Mehrprozessorkonzept, in dem ein Prozessor die I/O-Last (FP VANESSA) abfängt und ein zentraler Prozessor (CP) die höheren Aufgaben für mehrere Kanäle erfüllt.

Diese Überlegungen führten zu folgendem Konzept:

Frontendprozessor VANESSA-2 mit zwei unabhängigen AX.25 Kanälen, den notwendigen Peripheriebausteinen und eine Z80-280 CPU von ZILOG.

IBM PC (XT oder AT kompatibel) als Routingprozessor (CP) für mehrere Frontendprozessoren.

DUAL-PORT-RAM als gemeinsames Interface zwischen dem CP und FP.

**Warum ein PC als Routingprozessor ?**

Der Einsatz eines PC als Routingprozessor bietet viele Vorteile. Betrachten wir einen Knoten mittlerer Größe, so muß zumindest ein Gehäuse und eine Stromversorgung für den Digitalteil (5 VDC) beschafft werden. Für Tests ist temporär eine Eingabeeinheit und ein Bildschirm notwendig.

Daneben sollten bei einem komfortablen Knoten Helptexte für die USER, Parameter, Konfigurationstabellen ect. gespeichert werden. Auch bei einem Stromausfall sollten diese Daten nicht verloren gehen, so daß der SYSOP die Daten nicht jedesmal neu eingeben respektive neu laden muß. Dazu bietet sich ein PC mit einem Floppy-Laufwerk geradezu an. Die Daten werden beim Start von der Floppy ins RAM gelesen und stehen dann für die USER zur Verfügung.

Betrachten wir die Kostenseite, so sind die Aufwendungen für die Beschaffung eines (gebrauchten) PC sehr bescheiden, für einen mittleren Knoten genügt ein 286 Typ, kleinere Knoten benötigen einen XT oder AT mit 10 MHz. Eine Harddisk ist nicht notwendig, eine Floppy von 720 KByte genügt vollkommen.

**2. Spezifikation:**

Der Frontendprozessor ist auf einer IBM-PC-Steckkarte aufgebaut und unterstützt zwei AX.25 Datenkanäle mit hoher Geschwindigkeit.



CPU		
Typ		ZILOG Z80 280,
Adressraum		64 KByte aus 16 MByte,
Run-Modi		System / User,
Cache		256 Byte,
MMU		eingebaut,
DMA		4 Kanäle,
Clock		9,8 MHz, 19,6 MHz optional,
Timer		3 getrennte Timer mit je 16 Bit.
Memory		
RAM		32 KByte, 128 KByte optional,
EPROM		32 KByte, 128 KByte optional,
DP-RAM		8 KByte.
Interface		
AX.25 Daten		ZILOG SCC 85C30, mit DMA,
RS-232C		Diagnose Terminal,
Opt. Anzeigen		5 LED,
Modem		2 Stück. AM-7911 1200bps,
Externe Modems		RS-422 Treiber,
Anschluß		Sub-D 37 poliger Stecker.
Datenrate		
interne Modem		2 x 1200 bps Full Duplex,
externe Modem		2 x > 38400 bps Full Duplex.

### **2.1 Prozessor:**

Als CPU wird der ZILOG Prozessor Z80-280 eingesetzt. Er ist binärkompatibel zum bekannten Z80 Prozessor, erlaubt jedoch durch seinen erweiterten Befehlssatz eine effiziente Speicherverwaltung und erspart durch die integrierten Timer DMA und UART einige externe Bausteine.

Die wichtigsten Merkmale sind:

Getrennte Stack - und Speicherbereiche für SYSTEM, USER, DATEN und PROGRAMMCODE,

Virtueller Speicherbereich von 16 MByte,

On-Chip-Peripherals: vier DMA Kanäle, drei 16 Bit Zähler, ein UART sowie Cache Memory.

### **2.2 Speicher:**

Der verwaltbare Speicherbereich umfaßt 16 MByte. Die eingebaute MMU ermöglicht je nach Programmierung bis 4 x 64 KByte Speicher im Direktzugriff zu benutzen. Die minimale Konfiguration umfaßt je 32 KByte EPROM und RAM für Code respektive Daten. RAM und EPROM können je bis zu 128 KByte ausgebaut werden. Für zukünftige Speichererweiterungen wurden 32 Pin Byte-Wide-Sockel vorgesehen. Zusätzlich ist der Einsatz von 8 KByte Dual-Port-RAM [DP-RAM] vorgesehen. Ein Arbiter verhindert Zugriffskonflikte. Bei gleichzeitigem Zugriff beider Prozessoren auf das DP-RAM wird der eine Bus mit einem Wait State blockiert. Die maximale Verzögerung ist gegeben durch die Zykluszeit der CPU, die gerade einen Zugriff durchführt, und beträgt maximal 70 ns.

### **2.3 Memory Decoder:**

Der Memory-Decoder fällt einfach aus, da durch den großen Speicherbereich die oberen Adressleitungen zu Decodierzwecken herangezogen werden können. Die Programmierung der MMU (Memory Management Unit) erlaubt eine beliebige Anordnung der physikalischen Speicherblöcke innerhalb der CPU Adreßsegmente und somit die vollständige Auffüllung des Adreßraumes im Direktzugriff mit Speicher.

### **2.4 Watchdog Timer:**

Ein Watchdog Timer soll die Zuverlässigkeit gegenüber Hardwareproblemen erhöhen. Keinesfalls sollen damit Unzulänglichkeiten der Software überspielt werden. Der Watchdog wird innerhalb der Timer-Interrupt-Routine neu getriggert, so daß ein unzulässiger Programmzyklus einen Restart der Z80-280 CPU auslöst. Falls VANESSA je in einem Satellit auf intergalaktischen Flügen eingesetzt werden sollte, ist es beruhigend zu wissen, daß ein Fehler vom Watchdog-Timer erkannt und zum Systemrestart führen wird.

## **2.5 Status Anzeigen:**

Zur Anzeige des Systemzustandes sind fünf LED Anzeigen vorhanden.

## **2.6 Modem:**

Als Modem wird der AM-7911 eingesetzt. Dieses Single-Chip Modem erlaubt eine Baudrate von 1200 bps gemäß BEL-202 Norm. Für größere Baudraten werden alle notwendigen Signale über RS-422 Treiber auf ein externes Modem zu führen.

## **2.6 RS-422 Treiber:**

Die Signale vom und zum Modem werden ausschließlich über RS-422 Leitungstreiber geführt. Dadurch kann einerseits eine praktisch beliebige Leitungslänge erreicht werden, andererseits werden durch die paarverdrillten Leitungen Einstreuungen und Erdschleifen vermieden.

## **2.7 I/O Funktionen:**

Für die Steuerung der Peripherie wurden zwei Output-Ports realisiert.

## **2.8 RS-232 Serial Port:**

Für die Inbetriebnahme eines PR-Knotens, Hardwaretests sowie Weiterentwicklung von Hard - und Software ist am Datenstecker eine serielle Schnittstelle RS-232 für ein Diagnoseterminal vorhanden. Über dieses Terminal können interne Testprogramme aufgerufen sowie Betriebsparameter überwacht und geändert werden. Für den Normalbetrieb ist kein Terminal notwendig.

## **2.9 Bemerkungen zum Hardwarekonzept:**

Durch die parallele Datenverarbeitung auf den Bus des Rechners hat es einen mächtigen Ruck im Datendurchsatz gegeben. Die Verarbeitungsgeschwindigkeit des Knotens wird nun durch seine Links (Baudrate und das Übertragungsverfahren (Semi - oder Voll duplex)) bestimmt. Die Laufzeiten innerhalb des Knotens sind durch die parallele Verarbeitung der Daten so gering geworden, daß sie auch bei einem 64 Kbit/s Voll duplexlink nicht ins Gewicht fallen. Beobachtungen des Knotens DB0NHM (der bereits jetzt von DB0EAM über eine Vanessakarte versorgt wird) zeigen deutlich, daß die Antwortzeiten, die der RMNC-Knoten ermittelt, nur ca. 50 % von seinem RMNC-Nachbarn DB0NID betragen (beide Funkstrecken 9600 Bit/s Semiduplex).

## **DUAL PORT RAM (DP-RAM):**

Bei konventionellen Lösungen (TheNet, TheNetNode) geht der ganze Datenverkehr vom Host-Prozessor zum TNC und umgekehrt über die serielle RS232-Leitung. Sind nun mehrere Kanäle angeschlossen, konzentriert sich das ganze auf eine oder zwei Schnittstellen. Inzwischen sind Bitraten von 9600 bps pro Link keine Seltenheit mehr, so daß diese Schnittstelle der eigentliche „Flaschenhals“ im System wird. Dazu kommt, daß serielle Schnittstellenkarten den PC - Bus stark belasten. Als Alternative besitzt die VANESSA ein DUAL PORT RAM. Damit ist ein schneller und einfacher Datentransfer vom und zum PC gewährleistet. Der PC und die VANESSA CPU können quasi gleichzeitig auf den Speicher zugreifen und die Übertragung erfolgt blockweise.

## **RS 422 Leitungstreiber:**

An einem Knoten sind komplexe Erdverhältnisse vorhanden. Verschleppung von Kriechströmen führt unweigerlich zur Verschlechterung der Datenübertragung. Neben der Antennenerde (Blitzschutz) sind noch Schutz Erde (Netzteil) und Ground der digitalen Signale zu verknüpfen, ohne daß Erdschleifen auftreten. Modemschaltungen für schnelle Links müssen möglichst dicht beim Modulator / Demodulator angeordnet sein, da sonst Einstreuungen und Leitungskapazitäten ihr Vorhandensein bemerkbar machen. Das bedeutet, daß einerseits der Aufbau des Modems möglichst dicht beim TRX (max. 30 cm Leitungslänge), andererseits der geordnete Aufbau in einem Rack verlangt wird. Zur Entkopplung sind an der VANESSA alle Signale zum oder vom Modem über RS 422 Leitungstreiber und paarverdrillte Leitungen geführt. Damit sind Leitungslängen von mehr als 500 m mit einfachem Signalkabel möglich, das System ist entkoppelt.

## **2.10 VANESSA Firmware:**

KISS MODE (Keep It. Simple and Stupid):

Für den Betrieb der VANESSA als TNC-Ersatz beim Knoten wurde eine KISS-Mode-Firmware entwickelt. Dabei übernimmt die VANESSA die Daten aus dem DP RAM paketweise und gibt sie über den SCC 85C30 (HDLC Controller) an die Modems weiter. Die Sendersteuerung (Slottime, DCD/Persistence Algorithmus, TX Delay ect.) erfolgt unabhängig vom Hostrechner durch die VANESSA. Empfangene Daten werden nach einem Plausibilitätstest via das DP RAM an den Hostrechner zur Verarbeitung weitergereicht. Damit ist der Hostrechner von den zeitkritischen Aufgaben weitgehend entlastet. Neben den eigentlichen KISS Routinen sind noch verschiedene Testroutinen für Senderabgleich und Hardwaretest (VANESSA Peripherie, Speicher LED ect.) vorhanden, die bei Bedarf von einem externen Terminal aufgerufen werden können.

## **HOST MODE:**

Im Gegensatz zum KISS-Mode übernimmt im Host-Mode der FP VANESSA die komplette AX.25-Layer2-Protokollsteuerung. Zwischen Host und VANESSA werden nur die Infodaten sowie der Status der Links (Connected, Info, Busy, Disconnected ect.) übertragen. Der Hostrechner besitzt die Masterfunktion, d.h. der FP wird aufgefordert, Daten oder Status an den CP zu senden.

Damit ist die Verwendung der VANESSA als Hochleistungs-TNC für DieBox oder Terminalprogramme wie SP, GP oder THP möglich. Bei Bedarf werden bis zu 30 Kanäle pro VANESSA unterstützt.

Der weiteren Entwicklung für PR oder verwandte Betriebsarten sind (noch) keine Grenzen gesetzt. Die Rechenzeitreserven sind noch nicht ausgeschöpft, Speicherplatz ist ebenfalls genügend vorhanden. Dank der universellen RS 422 Schnittstellen kann jedes beliebige Modem mit internem oder externem Clock adaptiert werden.

### **3 Installation:**

Der Aufbau der VANESSA Hardware wurde bewußt universell ausgelegt, um individuelle Konfigurationen zu ermöglichen. Die einzelnen Funktionen sind aus dem Schema ersichtlich, die wichtigsten werden hier kurz erklärt.

#### **3.1 Grundkonfiguration Jumper:**

2 x 1200 bps, BEL 202, Port 0/1. (Vanessakarte so gehalten, daß die SUB-D 37 nach rechts zeigt.)

X1 links	X2 links	X23 links	X21 links
X18 oben	X17 unten	X16 unten	X15 oben
X14 unten	X9 unten	X10 unten	X11 oben
X12 oben	X13 oben	X8 unten	X7 oben
X6 oben	X5 unten	X4 unten	

X24, X25, X28, X29, X27 und X26 bleiben geschlossen.

DIP Schalter SW 1 alle Switches auf ON.

IC 13 und IC 14 bestückt (Modem AM-7911). IC 33 bis 36 leer (RS422 Schnittstelle).

#### **3.2 Einstellung Port-Adresse:**

Die Port Adresse wird mit dem DIP Switch SW 1 definiert. Damit wird die Adresse des DUAL PORT RAM festgelegt.  
x dont care

x	ON	ON	ON		Port Nr. 0/1	Karte 1
x	OFF	ON	ON		Port Nr. 2/3	Karte 2
x	ON	OFF	ON		Port Nr. 4/5	Karte 3
x	OFF	OFF	ON		Port Nr. 6/7	Karte 4
x	ON	ON	OFF	*	Port Nr. 8/9	Karte 5
x	OFF	ON	OFF	*	Port Nr. 10/11	Karte 6
x	ON	OFF	OFF	*	Port Nr. 12/13	Karte 7
x	OFF	OFF	OFF	*	Port Nr. 14/15	Karte 8

Bemerkung:

Vanessa Port 8-15: Da es für die Adressierung der LED Adreßkonflikte mit einigen HD-Controllern gab, wurden die Adressen verschoben. Dadurch muß leider eine Hardwareänderung auf der Vanessakarte durchgeführt werden. Am IC29 muß der Pin 16 (bei Port 0-7 auf +5 Volt) auf GND umgelegt werden. Das kann am einfachsten mit einem Zwischensockel und einer Drahtbrücke gemacht werden, z.B. nach Pin 18.

#### **3.3 Verwendung Modem nach G3RUH:**

Bei Einsatz von Modems nach G3RUH entfallen IC13 (Port A) und IC14 (Port B). Die Jumper X24 und X28 (Port A) resp. X25 und X29 (Port B) sind offen. Anstelle der internen Modembausteine AM-7911 werden die entsprechenden Leitungstreiber IC33 bis IC36 gesteckt.

Achtung: CTSa (b) und RTXC a (b) (IC35/36 Pin 11 und 13) werden vom G3RUH Modem nicht unterstützt, und DÜRFEN NICHT in die Sockel gesteckt werden !

Konfiguration Jumper:

Port A	Port B	X24	X25	X28	X28
G3RUH	1200 bps	OFF	ON	OFF	ON
1200 bps	G3RUH	ON	OFF	ON	OFF
G3RUH	G3RUH	OFF	OFF	OFF	OFF

Die Vanessakarte stellt an ihrem Ausgang TRXCa (b) den 32-fachen Takt der eingestellten Baudrate zur Verfügung. Für das G3RUH-Original Modem wird jedoch nur der 16-fache Takt benötigt. Deshalb muß dieser noch mal durch 2 geteilt werden, z.B. mit einem 74HC74, was der Flankenqualität zugute kommt.

### **3.4 Modem mit eigenem Clock Generator:**

Die DF9IC-Modem bieten im Gegensatz zu den Original-G3RUH-Modem Ausgänge für den RX - Takt und den TX - Takt. D.h. das Modem erzeugt die entsprechenden Takte intern.

Zum besseren Verständnis sollte man die VANESSA - Pläne, Seite 5 und Seite 2, zur Hand haben.

Die Beschreibung bezieht sich nur auf Port A der VANESSA, der Port B wird ähnlich verdrahtet.

Der RX - Takt wird an IC35 Pin 14/15 (RS422-Seite) gelegt und gelangt über Pin 13 dann auf den - RTXCa .

Für den TX - Takt muß ein 26LS32 als Eingang mißbraucht werden, es bietet sich für Port A IC35 Pin 9/10 an. Der Pin 11 von IC35 darf nicht mehr in den Sockel gesteckt werden, sondern wird mit einer kleinen Drahtbrücke auf den Anschluß - TRXCa an IC34 gelegt. Der entspr. Treiber auf IC34 sollte durch herausbiegen des IC - Beinchen außer Betrieb genommen werden (Verbraucht sonst nur Strom). Werden auf beiden Vanessa - Ports DF9IC-Modem eingesetzt, kann IC34 auch komplett eingespart werden.

Da der Anschluß - CTsA jetzt offen liegt, muß dieser auf Masse Potential gelegt werden !

Zum Schluß muß noch der Taktteiler IC27 Port A bzw. IC32 Port B von der Vanessa entfernt werden. Da die VANESSA bei externen Takt - Modus **NUR** NRZ - Codierung verarbeiten kann, müssen die Modems mit NRZ - Gal bestückt sein. (Müssen extra bestellt werden !)

Noch eine Warnung zum Schluß:

Verwendet unbedingt die RS422-Schnittstellen, ohne die Treiber gibt's bereits bei 20 cm Kabellänge bei 38k4 schon erheblich Probleme und der SCC wird durch Potential - Unterschiede unnötig gefährdet !

Parametereinstellung:

Als Parameter muß im Mode dann ein „e“ wie Externer Takt angefügt werden. Für einen 38k4 Vollduplex - Link währe der richtige Eintrag „mode=38400de“. Die Baudratenangabe ist zwar dann unwichtig, da der SCC auf der VANESSA ja vom Modem getaktet wird, aber die richtige Baudrate ist informativer.

### **3.5 Test der Konfiguration:**

Für den Funktionstest der VANESSA wird ein Terminal am 37-Sub-D-Stecker angeschossen. Der RS232-Anschluß wird mit 4800 Bd., 8 Bit, no parity betrieben.

Eine Beschreibung der FIRMWARE-Testschritte ist im VANESSA USER Manual enthalten. Beim Start des PC wird auf der RS 232 Schnittstelle ein Logon Menü ausgegeben, die beiden unteren LED erlöschen.

Das Interface VANESSA zum PC kann nun mit dem Hilfsprogramm VANUTIL.EXE ausgetestet werden. Nachfolgend eine kurze Erklärung der Testschritte:

1. Show Konfiguration  
Anzeige der eingestellten Adresse des DP-RAM und der FIRMWARE Version.
2. PC I/O Port Test  
Test der beiden oberen LED an der Steckerplatte.
3. Dual Port RAM Test  
Statischer Test des Dual-Port-Ram. Dabei wird ein Testpattern eingelesen und anschließend verglichen.
4. Vanessa Reset  
Test der RESET-Leitung zur VANESSA, die unteren LED werden ebenfalls angesteuert.
5. Dynamic RAM Test  
Dies ist der wichtigste Test! Dabei werden Daten ins Dual-Port-Ram geschrieben, durch die Vanessa an eine andere Adresse verschoben und durch den PC wieder ausgelesen und verglichen. Dabei wird das Timing auf dem PC-Bus sowie die korrekte Funktion des Dual-Port-Ram geprüft. Treten hier Fehler auf, so besteht die Gefahr, daß Daten korrupt übertragen werden. Hinweise zur Fehlersuche im folgenden Abschnitt.

### **3.6 Dual Port Mode:**

Mit Hilfe des Programmes „VANCONF.EXE“ lassen sich leicht Multi-Baud-Userzugänge einrichten. Die beiden verknüpften Ports müssen jedoch auf derselben Karte liegen. Unter Dual-Port-Mode werden die PTT und DCD der beiden Ports gegeneinander softwaremäßig verriegelt. Der Dual-Port-Mode muß beim Starten des Rechners mit Hilfe der BAT-Datei bereits festgelegt werden. Soll der Port 0 + 1 verwendet werden, so ist das Programm „VANCONF 0 D“ aufzurufen. 0 = Port 0 + 1 D = Dual-Port-Mode. Die Baudraten brauchen nicht gesetzt zu werden, da sie durch TNN eingestellt werden.

### **4. Probleme, Fehlersuche:**

VANUTIL.EXE erkennt keine Vanessa-Fehler im Dual-Port-RAM oder Adressierlogik zum PC-Bus. VANESSA-Firmware, CPU und RAM kontrollieren mit DEBUG und Terminal an der RS232. Erlöschen die beiden unteren LED beim Start ? Kontrolle der Adreßmodifikation bei Port Nummer 8 und größer. Fehler im Dual-Port-RAM Bereich können mit dem DEBUG-Programm ebenfalls verifiziert werden. Die Adresse des Dual-Port-Ram kann wie folgt berechnet werden: Startadresse: 0xD000:0000 + (n \* 0x1000), d.h. 0xD000:0 beim ersten Port, D100:000 beim zweiten Port usw. In diesem Bereich (0xD000:0 - 0xD000:fff können mit DEBUG Daten eingegeben und wieder ausgelesen werden (n = Port Nr. 0 ...15).

Datenverluste:

Mit einem schnellen Rechner wurden verstümmelte Frames oder gar Datenverlust an der Vanessa-Schnittstelle beobachtet. Mit einem XT-Board waren keine Probleme zu beobachten, aber ab ca. 16 MHz traten diese Störungen auf. Das Dual-Port-Ram auf der Vanessa muß deutlich schneller als 100 ns sein, mit 70ns-Rams konnten bei 33 MHz Rechnern keine Probleme mehr beobachtet werden. Bei 386er oder 486er sollte das Mainboard-Setup auf folgende Werte gesetzt werden, da der AT-Bus-Clock mit 8 MHz spezifiziert ist:

- DRAM READ 1 WAIT STATE,
- DRAM WRITE 1 WS,
- EXTRA AT CYCLE WS 1,
- AT BUS CLOCK CLK / x (so daß der Bus-Clock ca. 8 MHz ist).

Versuche auf dem DB0EAM-Rechner haben ergeben, daß die Vanessakarten auch bei 12 MHz Bus-Clock mit VANUTIL8.EXE keinerlei Fehler zeigten.

Alternativ kann die Bestückung der VANESSA geändert werden. Beim Einsatz von ALS-Typen anstelle der HC- oder HCT-Typen im Bereich des Dual-Port-RAM konnten auch bei schnellerem Bus-Clock keine Fehlfunktionen mehr festgestellt werden.

Anstelle der HC/HCT-Typen werden IC 20, 21, 22, 23 24, 25 und IC 28 gegen ALS-Typen ausgetauscht.

#### **LED**

Zu den LED kommen auch immer wieder Fragen. An der Sub-D-37-Leiste sitzen die grünen LED. Sie werden von dem Z280 Prozessor gesteuert und zeigen das Senden eines Frames an (nicht zu verwechseln mit der PTT, die bei Duplex-Betrieb ja für 3 Minuten nach dem letzten Frame getastet bleibt). Diese LED werden deshalb auch beim VANUTIL-TEST nicht angesteuert. Ist externer Clock bei der Vanessa eingestellt, aber KEIN Modem angeschlossen, so können diese LED leuchten, da der SCC dann einen Takt erwartet.

Die darüberliegenden gelben LED zeigen den Empfang von Daten via dem Dual-Port-RAM, also über den Rechnerbus, an. Die Funktion der LED läßt sich mit dem VANUTIL-Test prüfen.

Die einzelne rote LED zeigt die Funktion des karteneigenen Prozessors an. Sie leuchtet, wenn der Prozessor steht (was eigentlich nie passieren darf..).

Die beiden linken LED (gelb und grün) sind für den Port A der Karte zuständig und somit die rechten beiden für den Port B.

Die oberen LED können auch mit dem Debug-Programm via PC-Port gesteuert werden. Die Adreßberechnung ist wie folgt: Linke LED Adresse 0x306 + (n \* 0x8), also 0x306, 0x30E ... Rechte LED 0x307 + (n \* 0x8), also 0x307, 0x30F usw.

Daraus ergibt sich die folgende Adressenliste:

Port 0 :	o 306 00	--> LED leuchtet (oben links)	o 306 01 --> LED aus
Port 1 :	o 307 00	--> LED leuchtet (oben rechts ...)	
Port 2 :	o 30e 00		
Port 3 :	o 30f 00		
Port 4 :	o 316 00		
Port 5 :	o 317 00		
Port 6 :	o 31e 00		
Port 7 :	o 31f 00		
Port 8 :	o 126 00	..Ist die Adresse nicht geändert:	o 326 00
Port 9 :	o 127 00		o 327 00
Port 10 :	o 12e 00		o 32e 00
Port 11 :	o 12f 00		o 32f.00
Port 12 :	o 136 00		o 336.00
Port 13 :	o 137 00		o 337 00
Port 14 :	o 13e 00		o 33e 00
Port 15 :	o 13f 00		o 33f 00

Die unteren beiden LED lassen sich genauso ansprechen unter den Adressen:

Port 0 + 1	o 300 00	--> LED Leuchten	o 300 01 LED aus
Port 2 + 3	o 308 00		
Port 4 + 5	o 310 00		
Port 6 + 7	o 318 00		
Port 8 + 9	o 120 00	..Adresse nicht geändert:	o 320 00
Port 10 + 11	o 128 00		o 328.00
Port 12 + 13	o 130 00		o 330 00
Port 14 + 15	o 138 00		o 338 00

**Offene Gatter auf der Vanessakarte:** Offene C-MOS Eingänge an Gattern neigen zum Schwingen. Deshalb sollten sie besser auf einem logischen Pegel liegen. Betroffen sind die folgenden IC: (Klammerwerte = IC-Eingang)

IC 9 (13); IC 20 (9-10) und (12-13); IC 25 (9-10) und (12-13);  
IC 26 (10) und (12) und IC 12 (4-5), (9-10) und (12-13).

SWISS ARTG  
Swiss Amateur Radio Teleprinter Group  
Geschäftsstelle Arturo Dietler, HB9MIR,  
Blauenweg 8  
CH-4335 Laufenburg  
Autor: HB9PAE @ HB9AJ

ANHANG D:**(Ethernet) ...Was muß ich tun ?**

Ethernet gibt es auf 2 unterschiedlichen Kabeln.

Zum einem ist der Betrieb auf dem uns sehr bekannten RG58 möglich. Hierbei MUSS man unbedingt beachten, daß BEIDE Enden des Kabels mit je einem 50 Ohm Abschlußwiderstand versehen werden müssen.

Das Gegenstück zum RG58 ist Twisted-Pair . Es besteht aus einer Doppelader und benötigt keinen Leitungsabschluß.

Wie geht man nun vor, wenn man bei Ethernet noch unsicher ist ?

- 1.) Zuerst muß man sich auf seinen Rechner einen freien IRQ suchen, den die Ethernet-Karte belegen kann.
- 2.) Dann ist der Adreßbereich der Ethernet-Karte im Bereich 0x2e0 - 0x320 festzulegen. Hier kann es zu Überschneidungen mit den Vanessakarten kommen, bitte beachten.  
**Die Vanessakarten benutzen die Adressen 0x300...0x33f (je nach Anzahl der Vanessakarten und verwendetem Port.) Außerdem wird default der Adreßbereich D000..Dxxx benutzt. Auch dieser muß bei gleichzeitiger Verwendung von Vanessakarten und Ethernet-Karten bedacht werden.**
- 3.) Nach dem Einbau in den Rechner kann nun die Ethernet-Karte auf Brauchbarkeit und Funktion mit dem Programm TESTDRV.EXE überprüft werden. Dazu muß der entsprechende Kartentreiber noch aufgerufen werden. Z.B.: NE2000.COM mit Angabe des Soft/Hardware IRQ und Adreßbereich der Karte. NE2000 0x60 0x300.
- 4.) Meldet nun TESTDRV keine Probleme auf den Rechnern, so kann man auf dem einen Rechner SEEPKT.EXE starten (dieses ist ein Ethernet-Monitor) und auf dem anderen Rechner SENDPKT.EXE (dieses ist eine Bake, die ständig Daten sendet). Die Daten, die SENDPKT.EXE erzeugt, sollen mit SEEPKT.EXE lesbar sein. Dann steht eigentlich einer Verbindung der 2 Rechner über Ethernet nichts mehr im Wege, sofern das in beiden Richtungen funktioniert.

Aber erst mal zum Testen der Schnittstellenkarte:

**TESTDRV - Ein Utility zum Testen von FTP-Paket Treibern:****Vorwort:**

Leider halten nicht alle mit den Netzwerkkarten gelieferten Treiber, was das Handbuch verspricht. Dies mag zum Teil an Mehrdeutigkeiten in den Spezifikationen liegen. Teilweise liegt es aber mit Sicherheit an falsch verstandenem Schutz vor „Datenspionen“ im Netz. Und einiges können wir auch auf das Konto „Schlampereien“ buchen.

Man kann nun den Treiber mit der Anwendungssoftware testen. Das ist oft unbequem und die Fehlermeldungen sind sparsam oder nichtssagend - wenn überhaupt welche kommen. TESTDRV soll hier für etwas Bequemlichkeit sorgen.

Wenn der mit der Karte gelieferte Treiber nicht richtig will, kann man bei den Standardkarten immer die Treiber aus dem FTP-Paket nehmen. Für NE2000 und WD (SMC) kompatible Karten sind Treiber in diesem Paket enthalten.

FTP-Paket Treiber sind m.W. die einzigen Treiber, die verschiedene Netzwerkprotokolle auf einer Karte gleichzeitig gestatten. Hinzu kommt, daß die Schnittstelle hervorragend dokumentiert ist und die gesamte Software dafür im Quelltext frei verfügbar ist.

In dieser Anleitung werde ich nicht viel über Netzwerkgrundlagen schreiben. Das soll in einer getrennten Dokumentation geschehen.

**Voraussetzungen:****Hardware**

Netzwerkkarten bekommt man heute „NE2000“ kompatibel für weniger als DM 50.-. Üblich ist sind „Cheapernet“ = BNC Anschluß (10Base2) und/oder UTP (RJ45) Anschluß (Twisted-Pair). Der BNC Anschluß hat den Vorteil, daß man ohne weitere Hardware einige Rechner koppeln kann. Wer einen Rechner „solo“ testen will, beachte bitte, daß der Netzwerkanschluß an beiden Seiten mit jeweils 50 Ohm abgeschlossen sein muß. Sonst verweigern die meisten Karten die Sendung. Also ein T-Stück mit zwei 50 Ohm Abschlüssen aufstecken.

Moderne Karten werden nicht mehr über Jumper konfiguriert, sondern per Software. Das kann Probleme machen, wenn die vom Kartenhersteller gewählten Defaults im Rechner bereits belegt sind. Dann muß man u.U. erst andere Karten ausbauen, um die Netzwerkkarte umkonfigurieren zu können. Eine NE2000 kompatible Karte braucht einen freien Interrupt und eine freie IO-Adresse. **ACHTUNG: Werden Vanessakarten eingesetzt, so sind die IO-Adressen je nach Port zwischen 0300-0338 bereits belegt !!**

**Software**

Wer schon einen Netzwerkanschluß an seinem Rechner hat, bedenke bitte, daß eine Netzwerkkarte nicht zwei Herren gleichzeitig dienen kann. Man muß also die vorhandene Netzwerksoftware (Novell, WfW usw.) totlegen, um mit FTP-Packet Treibern experimentieren zu können. Anschließend kann man dann versuchen, die „alte“ Software auf der Basis des Packet-Treiber zu installieren. Nur in dieser Richtung geht es, nicht anders herum. Und ein Packet-Treiber braucht nie mehr als einen Aufruf zum Installieren (Beispiel „EP3000PD 0x60“). PROTMAN.SYS und ähnliche Programme sind hier definitiv falsch.

Bei der Installation muß i.A. ein Software-Interrupt für den Treiber angegeben werden. Dieser Interrupt kann per Definition im Bereich 0x60 bis 0x80 liegen. Es spricht nichts dagegen, 0x60 zu verwenden.

**Testablauf**

Als erstes wird die Signatur des Treibers gesucht und so der verwendete Software-Interrupt gefunden. Natürlich könnte man auch den Software-Interrupt im Aufruf angeben. Da aber fast alle Programme den Treiber über seine Signatur suchen, macht es schon Sinn, hier auch so vorzugehen.

Dann müssen wir dem Treiber mitteilen, was wir von ihm verlangen - den Netzwerktyp, welche Art Pakete usw. Außerdem müssen wir ihm sagen, wo er die Pakete abliefern soll, die „für uns“ eingegangen sind. Der Treiber liefert uns als Ergebnis ein „Handle“, das zukünftig bei der Kommunikation zwischen dem Treiber und uns den Briefkasten bezeichnet. Ohne gültiges Handle wird nicht gearbeitet.

Nun fragen wir den Treiber, welchen Befehlssatz er versteht. Es gibt drei Ausbaustufen: Basisdienste, erweiterte Dienste und highperformance Funktionen. Wir brauchen die erweiterten Dienste. Die ersten Versionen von TFNETR und TNN liefen auch mit den Basisdiensten. Um zu PE1CHL und G8BPQ kompatibel zu werden, haben wir das Protokoll aber noch einmal geändert.

Danach fragen wir die Hardwareadresse der Karte ab. Diese Adresse ist eindeutig, jede Karte hat eine andere Adresse. Wir sollten sie nie ohne zwingende Gründe ändern. Bei dieser Abfrage erleben wir die Schönheiten der Segmentierung des 80X86 Prozessors. Wenn der Programmierer des Kartentreibers nicht sorgfältig war, kommt hier als Adresse 00:00:00:00:00:00. Und dann sind wir ziemlich sicher, daß die gleiche Schlamperei auch in der Empfangsroutine drin ist. Damit sind dann Abstürze vorprogrammiert.

Die Kommunikation zwischen zwei Netzwerkkarten kann auf mehrere Arten erfolgen:

- Im Paket werden die Hardwareadressen der beteiligten Karten angegeben. Das ist kompliziert. Denn ich muß vor der ersten Kontaktaufnahme die Adresse meines Partners kennen.
- Die Pakete werden „an Alle“ gesendet (Broadcast). Für diese Zwecke ist die Zieladresse FF:FF:FF:FF:FF:FF definiert. Dieses Verfahren stört in größeren Netzen sehr, weil wir dann mit unseren Paketen das gesamte Netz überschwemmen und jeder unsere Pakete hören und analysieren muß.
- Die Pakete werden an eine Gruppe gesendet (Multicast). Nur die Gruppe hört auf diese Pakete. Dieses Verfahren werden wir verwenden. Wir kontrollieren nun, ob unser Treiber dieses Verfahren empfangsseitig versteht.

Der nächste Test erfolgt, um zu sehen, ob das Programm NETWATCH laufen wird. NETWATCH ist ein universeller Netzwerkmonitor. Leider haben viele Programmierer den dazu notwendigen „promiscuous mode“ zugenagelt. Aber es gibt ja die FTP-Treiber.

Nun setzen wir den Treiber sendeseitig in den Multicast-Mode. Dazu definieren wir die „BPQ“-Adresse.

Als letzten Test schicken wir ein Paket (mit zufälligem Inhalt) an unsere Gruppe. Wenn der Abschlußwiderstand nicht steckt, bleiben einige Karten hier hängen. Andere kommen mit einer Fehlermeldung.

Zum Schluß geben wir das Handle wieder frei. Wenn der Test bis hierher durchgelaufen ist, besteht Hoffnung, daß der Treiber auch mit der Anwendersoftware laufen wird.

Ein letzter Hinweis zum Schluß: ein Rechnerabsturz beim Empfang oder Senden eines Paketes in Verbindung mit einer WD (SMC) Karte ist meist auf einen Fehler in der Konfiguration des Speichermanagers zurückzuführen. Diese Karten nutzen nämlich Teile des Hauptspeichers zur Kommunikation mit dem Treiber.

**Fehlermeldungen:**

Einige Fehlermeldungen sollten nie vorkommen. Sie sind mit einem ??? markiert. Treten sie dennoch auf, so deutet das auf einen kapitale Macke im Treiber hin.



kein Fehler ???

Der Treiber hat einen Fehler gemeldet und erklärt bei Nachfrage, es sei nichts gewesen.

ungültiges Handle ???

Der Treiber erinnert sich nicht an sein eigenes Handle.

ungültige Klasse

Der Treiber unterstützt die Klasse „ETHERNET“ nicht.

ungültiger Typ ???

Wir haben „irgendeinen“ Typ verlangt.

ungültige Nummer ???

Wir haben „die erste Karte“ verlangt.

ungültiger Packet Typ ???

Wir haben „alle verfügbaren“ Typen verlangt.

Multicast unzulässig

„Unser“ Modus wird nicht unterstützt, Treiber unbrauchbar.

TERMINATE unzulässig ???

Wir verlangen nicht, daß der Treiber sich entfernt.

unzulässiger RX-Modus

„Unser“ Modus wird nicht unterstützt, Treiber unbrauchbar.

nicht genug Speicher

Karte ist überfordert, nach RESET des Rechners noch einmal versuchen. Ist noch andere Netzwerksoftware aktiv?

Type ist belegt

Andere Software hat die Karte undefiniert hinterlassen.

unzulässiger Befehl

Ein Standardbefehl wird nicht erkannt, Treiber unbrauchbar.

(Hardware) Fehler beim Senden

Abschlußwiderstand nicht gesteckt oder falscher Anschluß.

kann Hardwareadresse nicht setzen ???

Wir fummeln da nicht dran.

falsches Hardware-Adressenformat ???

Der Treiber ist defekt.

RESET nicht möglich ???

Da wurde ein Befehl mißverstanden.

Auf der TNN-Seite ist nun folgendes nötig:

Es muß in der Start.bat ein Ethernetkarten-Treiber aufgerufen werden. Treiber einfach mal aufrufen, meist sagen sie dann, was sie gerne für Parameter hätten. Hierzu ist auf der Diskette ein DIR TESTDRVR mit ein paar Hilfsprogrammen.

```
tnll.exe -v
nwpd 0x68 10 0x340
ipxpd.exe /i=0x68

TNNGNU.exe

tnll.exe -u
nwpd -u
```

**ANHANG E:****DER PACSAT-BROADCAST-SERVER von TNN****Neues ab TNN V1.70:**

- Parameter komplett im BROADCAST-Server,
- DIRECTORY-BROADCAST für WiSp,
- Neue Nachrichtenverwaltung,
- Max. Anzahl gespeicherter Mail einstellbar.

**ABLAUF:**

TNN macht simplen DieBox-S&F mit einer Mailbox und speichert die Nachrichten auf der Platte und macht dann mit den Daten seinen Broadcast. Der Befehl „BOX“ schaltet TNN in den S&F-Modus. Die Nachrichten werden so, wie sie über den S&F kommen, mit aufsteigenden Nummern gespeichert. Beim Start von TNN werden die 1. und die letzte Nummer ermittelt.

**Der Broadcast:**

Das Protokoll ist in PACSAT.TXT dokumentiert, er ist 100% PACSAT kompatibel. Der PACSAT-Server benutzt optional extended header und überträgt im RLM Message Format. Damit sollte auch z.B. WISP (bei AMSAT zu bekommen) klarkommen und der User kriegt die Nachrichten automatisch in Rubriken sortiert. Ein von PE1HCL umgestrickter Headergenerator heißt jetzt PFHADD.EXE (V3.3 vom 2.6.95), er muß im PACSAT-Verzeichniss stehen. Damit WiSp mit dem TNN-PACSAT-BROADCAST zurecht kommt, wird vor dem eigentlichen File noch DIR-Eintrag (ab TNN V1.70) ausgesendet.

**Bedienung:**

- Nachrichteneingang durch DieBox-1.8-Kompatiblen S&F (keine BIN-Mail !).
- User&Sysop-Schnittstelle (Befehl PACSERV, PAC\*), dort stehen zur Verfügung:

**Für den Sysop:****+ Port DestCall via DestRoute**

Broadcast für einen Port aktivieren.

Beispiel: + 1 QST-1 via DB0LWL

DestCall sollte immer QST-1 sein. Broadcast können VIA L2 ausgestrahlt werden. Dies sollte man aber mit Vorsicht tun, da ein 9600-Baud-Broadcast via ein Nachbarknoten diesen in der Regel zum Absturz bringt, da dieser die riesige Datenmenge nicht schnell genug transportieren kann. Dies gilt insbesondere für RMNC/FlexNet/BayCom-Nachbarn.

**- Port**

Broadcast deaktivieren.

**R**

Reload, das Filesystem nach einer externen Veränderung neu einlesen.

**M**

PacSat-Boxcall setzen, M call, sollte das Rufzeichen des Knotens sein oder ein Fantasie-Call, es muß aber mit dem Rufzeichen übereinstimmen, mit dem man bei der Box eingetragen ist. Dieses Rufzeichen wird niemals HF-mäßig gesendet. Es wird eigentlich nur benötigt, da bei DieBox alle Partner unterschiedliche Calls haben müssen. DB0HRO-5 (Box) macht mit DB0HRO-7 (Cluster) S&F, somit kann man dort nicht mehr DB0HRO (Knoten) als S&F-Partner hinzufügen, also trägt man dort den Knoten als BR0CST (oder was auch immer) ein und stellt M auf M BR0CST. Dann sollte es klappen.

**P**

Zum Einstellen verschiedener Parameter

**Ablauf des Broadcast:**

TNN hat eine Prioritätsschaltung, welche die letzten 200 eingegangenen Nachrichten im 10-15 Minutentakt wiederholt, damit User nur kurze Zeit einzuschalten brauchen und sofort die wichtigsten Nachrichten haben. Für den Rest der Zeit werden die sonstigen Nachrichten ausgesendet.

**Broadcast Format:**

Ist 100% kompatibel zu dem, was unsere künstlichen Erdtrabanten machen, inkl. DIRECTORY-BROADCAST.

**Installation:**

In TNN176.PAS muß der PACSATPATH angefügt werden, er sollte auf C:\TNN\PACSAT\ stehen. Dort werden alle Nachrichten gespeichert. In dieses Verzeichnis muß auch die Datei „PFHADD.COM“ kopiert werden, sie wird intern benötigt. Dann muß man den Sysop der nächsten Mailbox bekniien, damit der S&F einträgt. Die Box muß, wenn sie den Knoten connected hat, „BOX“ eingeben, um in den S&F-Modus zu kommen. Sobald die 1. Nachricht eintrifft, geht das Broadgecaste los, man kann auch auf mehreren Ports gleichzeitig broadcasten (es geht auch auf einem mit 9k6, auf dem anderen mit 19k2, d.h. die Abfolgen sind unabhängig, der eine Port wartet nicht auf den anderen). Der Knoten benötigt (natürlich) eine Festplatte, sie sollte schon 80MB oder so haben, mehr ist auch nicht schlecht.

**SONSTIGES:****Der User:**

WISP ist ein Programm, das automatisch die Mail dekodiert und in Rubriken sortiert.

**Wartungsaufwand für den Sysop:**

Wenig. Wenn der Broadcast erstmal installiert ist, läuft er von selbst und braucht nicht gewartet zu werden. Wenn auf der Platte ein Lesefehler auftritt, kann der Sysop eine Freifahrt haben, das ist bei Boxen aber auch so, ist leider nicht zu ändern.

**Wichtig:**

TNN muß pro Sekunde etwa 1,5kb von der Platte lesen, deshalb ist SMARTDRV mit NUR-LESE-CACHE zu empfehlen, es schont die Platte. Der VORLESESPEICHER sollte von standardmäßig 8kb auf 32kb gestellt werden (damit mal experimentieren).

Vanessa ist Tokenring vorzuziehen, der Tokenring muß bei 9k6 Broadcast 19k2 haben, 38k4 ist besser, auf vollen Digi (viele Links, z.B. DB0FC) kann es passieren, daß Dauertastung nicht geht, das ist nicht zu beheben. Mit Vanessa sollte es nicht passieren.

**Doppelte-Nachrichten:**

Werden nicht überprüft. User, die über Box-Software verfügen (WinGT), die S&F kann, können auch in den Server einspielen, das geht dann auch auf dem Ausgang raus. Der Sysop sollte das ein wenig überwachen, damit die Leute keinen Unfug machen.

**Request:**

(Also anfordern von fehlenden Teilen durch UI-Frames) ist für unsere Verwendung nicht sinnvoll, wird es auch erstmal nicht geben.

**Spezial:**

In einigen Gebieten ist ein PR-BROADCAST-Ausgang im ISM-Bereich wegen vorhandener ATV-Belegung nicht möglich. Aus diesem Grund wird z.B. in Köln BROADCAST auf der Eingabe (!) eines FM-Relais gemacht. Dazu werden immer nur 2 Frames gesendet, um dann zu prüfen, ob das Relais geöffnet wurde. Ist das Relais noch geschlossen, so werden wieder 2 Frames gesendet. Wurde das Relais aber geöffnet, so wird über eine kleine Schaltung die PTT gesperrt. Der Durchsatz ist natürlich geringer als bei einem DAUER-TX-Ausgang, aber ich denke, daß es eine gute Alternative ist. Zu diesem Zweck habe ich eine kleine Schaltung entwickelt. Bei Fragen bitte wenden an: (DG1KWA / Andreas @DB0GSO).

**ANHANG F:****Aufbau der Layer 3 Frames bei TheNet, TheNetNode und TheBoxware.**

Da für die 3 Programme TheNet, TheNetNode und TheBoxware dasselbe Layer 3-Protokoll gilt, wird im folgenden nur von TheNet die Rede sein, aber es sind alle 3 Programme gemeint (wenn nicht ausdrücklich anders vermerkt). Hier werden nur die Frames mit erweitertem AX.25-Format behandelt, Identifikations-Frames etc. bleiben unberücksichtigt.

Alle speziellen Layer 3 Frames von TheNet werden mit dem PID „CF“ ausgesendet, wodurch eine Unterscheidung von normalen Layer 2 Frames gewährleistet ist (bei denen ist der PID „F0“). Dies gilt sowohl für I-Frames als auch für UI-Frames.

Das zusätzliche Protokoll im Infoblock des AX.25-I-Frames besteht grundsätzlich aus 3 Blöcken: der Netzwerk-Kontrollblock (Layer 3) besteht aus 15 Bytes, der darauf folgende Transport-Kontrollblock (Layer 4) besteht aus weiteren 5 Bytes, und schließlich bleibt noch der Informationsteil, für den eine Länge von 0 bis 236 Bytes vorgegeben ist.

**Netzwerk Layer Frames.**

Der Router von TheNet bestimmt, auf welchem Weg Nachrichten von einem Zielknoten zum anderen übertragen werden. Das Routing erfolgt automatisch auf dem vermutlich günstigsten Weg. Der während einer Verbindung verwendete Weg kann sich ändern, wenn z.B. ein Netzknoten ausfällt; dann wird automatisch ein alternativer Weg verwendet (wenn vorhanden) - der Enduser merkt von der Umleitung nichts.

Für das Routing bei TheNet werden zunächst Informationen über die erreichbaren Zielknoten mit einer Broadcast-Aussendung ausgetauscht. Bei TheNetNode wird die Laufzeit zum Nachbarknoten gemessen.

Der höhere Transport-Layer baut auf dem Netzwerk-Layer auf, so daß alle TheNet-I-Frames einen Netzwerk-Kontrollblock enthalten, der immer gleich aufgebaut ist. Er besteht aus insgesamt 15 Bytes. Als erstes kommt das Rufzeichen des Absenderknotens (7 Bytes), gefolgt vom Rufzeichen des Empfängerknotens (7 Bytes). Schließlich wird die Lifetime dieses Frames übertragen (1 Byte). Die Lifetime wird bei jeder Weitergabe an einen Nachbarknoten um 1 vermindert, bis entweder der Zielknoten erreicht ist, oder aber der Wert 0 wird. Falls der Wert 0 erreicht wird, ist das Frame zu vernichten, damit ein Frame nicht durch eine zufällig entstandene Schleife des Netzes endlos lange kreist und dadurch das Netz blockiert.

**Laufzeitmessung bei TheNetNode.**

Um die Laufzeit für Layer 3 Frames zu einem Nachbarknoten und zurück zu messen, werden von TheNetNode regelmäßig Meßframes an alle connecteten Nachbarknoten geschickt. Dabei handelt es sich um I-Frames mit dem PID „CF“. Im Netzwerk-Kontrollblock ist als Absender das Rufzeichen des Knotens eingetragen, der die Laufzeitmessung vornimmt. Als Zielknoten ist das Phantom-Rufzeichen L3RTT (wie „Layer 3 Round Trip Time“) eingetragen. Der Transport- Kontrollblock ist mit 5 Bytes „00“ besetzt. Im Informationsteil sind weitere Daten wie z.B. die Systemzeit beim Erzeugen des Frames enthalten. Da diese Daten nur beim Absenderknoten ausgewertet werden, ist das genaue Datenformat nicht wichtig. Außerdem ist dieser Teil des Protokolls noch in der Entwicklungsphase, so daß Änderungen hier nicht auszuschließen sind.

**Automatisches Routing (alter Art).**

Diese Art des Routings wird noch für Knoten bereitgestellt, die noch nicht das Protokoll von ON5ZS unterstützen.

Für das automatische Routing werden regelmäßig von den Netzknoten Informationen über den Inhalt der Zielknotenliste ausgesendet. Dies erfolgt durch UI-Frames mit dem TheNet-PID „CF“. Das Absender-Rufzeichen dieser Routingframes (Broadcast) ist das Rufzeichen des absendenden Netzknotens, und als Zielrufzeichen wird das Phantom-Rufzeichen „NODES“ verwendet. Für TheNetNode ist das Zielrufzeichen nicht mehr vom Programm vorgegeben, um auch über spezielle Layer 2 Digipeater (z.B. FlexNet) routen zu können.

Im Info-Feld des Broadcast-Frames wird zuerst als Kennung ein Byte „FF“ gesendet, gefolgt von dem Ident des Absenderknotens (6 Bytes). Schließlich folgen bis zu 11 Einträge der Zielknotenliste.

Jeder Zielknoteneintrag besteht aus 4 Datenblöcken. Der erste Datenblock enthält das Rufzeichen des Zielknotens (7 Bytes). Der zweite Datenblock beinhaltet den Ident des Zielknotens (6 Bytes). Im dritten Datenblock folgt der Nachbarknoten, über den der Zielknoten mit der besten Laufzeit bekannt ist (7 Bytes). Schließlich wird im vierten Datenblock die Laufzeit des besten Weges zum Zielknoten angegeben (1 Byte).

Alle angegebenen Idents werden als normale ASCII-Zeichen übertragen, während für alle Rufzeichen das AX.25-Format (ASCII um 1 Bit nach links geschoben) verwendet wird.

Bei neueren TheNet-Versionen ist die Aussendung der Zielknotenliste erweitert um die Layer 3 Lifetime der einzelnen Zielknoten. Damit diese Erweiterung kompatibel zu den älteren Versionen bleibt, wurde der zusätzliche Teil an die bisherige Aussendung angehängt. Als Kennung für den Beginn des Lifetimeblocks wird ein Byte „00“ gesendet. Diese Kennung verhindert gleichzeitig eine weitere Auswertung des Frames durch ältere Programme, da sonst statt der Lifetimes ein weiterer Zielknoten folgen müßte, beginnend mit einem Rufzeichen. Rufzeichen können aber nicht mit „00“ anfangen.

Darauf folgt die eigene Layer 3 Lifetime des Absenderknotens. Anschließend werden die Lifetimes der in dem Frame enthaltenen Zielknoten in der Reihenfolge ausgesendet, in der vorher die Zielknoten ausgesendet wurden.

Für die Lifetime wird jeweils 1 Byte gesendet und zwar diejenige Lifetime, die auch in einem Frame des entsprechenden Zielknotens eingetragen wäre. Ist die Lifetime unbekannt, wird statt dessen „00“ gesendet.

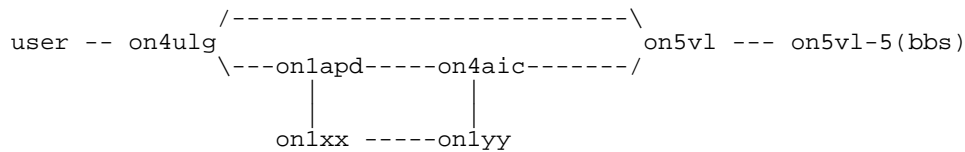
### Neues Routing in TheNetNode von ON5ZS:

Routeränderungen:

Der BROADCAST wird nun im sicheren Level 3 übertragen. So wie ein Node verschwindet, wird seine Qualität auf „1“ gesetzt. Die Nachbarn werden SOFORT durch den BROADCAST darüber verständigt. Diese verständigen auch sofort ihre andere Nachbarn und so weiter. Damit wird das gesamte Netz davon benachrichtigt. Dauer der Ausbreitung: zwischen 0 und 4 Sekunden pro Hop.

Ein neues System:

1 Strecke und 2 Wege. Das System ist jetzt soweit klar.



Die Lösung war ziemlich einfach...

Damit on5vl den Weg on5vl, on4aic, onlapd nimmt, muß bei on4aic der beste Weg nach on4ulg über onlapd eingetragen sein. Durch die Art der Berechnung der Qualität, die Qualität von on4aic bei on5vl ist gleich der Qualität von on5vl bei on4aic, kann on5vl sehr einfach die Qualität der Strecken von on4aic berechnen (und entscheiden, ob on4ulg direkt angesprochen wird oder nicht).

- a) Wenn es möglich ist (kein ping-pong -> Berechnung der Qualitäten des Nachbarn).
- b) Der alternative Weg wird gewählt, wenn die Übertragungszeit eines Frames auf der alternativen Strecke kürzer war.
- c) Wenn ein Frame vom Level 4 kommt (on4aic -> onl yy wird niemals gebraucht).

Die alternative Strecke wird beansprucht ab dem Augenblick, wo eine bestimmte Anzahl (\*) von Frames im tx\_buffer level2 (-> tosend) bei on5vl -> on4ulg vorhanden ist.

- (\*) Diese Anzahl verändert sich nach der Qualitätsdifferenz zwischen Hauptstrecke und alternativer Strecke in der Art, wie die Frames in etwa in der richtigen Reihenfolge bei on4ulg ankommen (TNN gibt sie in der richtigen Reihenfolge weiter).

Es gibt kein „Selective retransmission request“ von on4ulg.

### Transport Layer Frames.

Zwischen den Endknoten eines Circuit wird ein spezielles Übertragungsprotokoll verwendet, um eine transparente, fehlerfreie Datenübertragung zu ermöglichen. Dieses zusätzliche Protokoll wird im Transport-Kontrollblock und im Informationsteil des Frames übertragen.

Der Transport-Kontrollblock besteht aus 5 Bytes. Die Bedeutung der ersten 4 Bytes ist abhängig vom 5. Byte, dem Kontrollbyte. Das Kontrollbyte selbst besteht aus dem Opcode sowie 3 Flags.

Die Bits 0-2 im Kontrollbyte bilden den Opcode, aus dem die Frameart abzuleiten ist. Von den 8 möglichen Opcodes werden derzeit nur 6 verschiedene Möglichkeiten verwendet:

- 1 = Connect-Request,
- 2 = Connect-Acknowledge,
- 3 = Disconnect-Request,
- 4 = Disconnect-Acknowledge,
- 5 = Information-Transfer,
- 6 = Information-Acknowledge.

Die Bits 3 und 4 werden derzeit (noch) nicht verwendet, sie werden daher immer auf 0 gesetzt.

Bit 5 ist das More-Follows-Flag. Dieses Flag zeigt an, daß der Informationsteil auf mehrere Frames aufgeteilt wurde. Um auch Frames des Enduser mit mehr als 236 Byte Informationsteil transparent zu übertragen, werden solche Frames aufgeteilt. Im ersten Teil werden die maximal möglichen 236 Bytes Daten übertragen, im nächsten Frame folgt der Rest. Bei älteren Versionen von TheNet wurden Informationen des Knotens selbst (Userliste, Nodesliste) zur Übertragung innerhalb des Netzes teilweise auch mit wesentlich größerer Anzahl an Datenbytes übertragen. Dafür wurden dann entsprechend mehrere Frames mit gesetztem More-Follows-Flag nacheinander gesendet. Beim Endknoten wurde dann die Unterteilung in AX.25-gerechte 256-Byte-Frames vorgenommen.

Bit 6 heißt NAK-Flag. Dieses Bit wird nur bei den Frametypen Information-Transfer und Information-Acknowledge verwendet und zeigt an, daß ein Frame wiederholt werden soll.

Bit 7 ist das Choke-Flag. Dieses Flag wird bei den Frametype Information-Transfer, Information-Acknowledge und Connect-Acknowledge verwendet. Die unterschiedliche Bedeutung dieses Flags ist bei den entsprechenden Frametypen erläutert.

### **Connect-Request.**

Mit einem Connect-Request-Frame wird ein neuer Circuit angefordert. Die ersten beiden Bytes des Transport-Kontrollblocks spezifizieren den Circuit des Absenderknotens. Dabei gibt das erste Byte einen Index in der Circuit-Tabelle an (Circuit-Index). Das 2. Byte wird als zusätzliche Kennzeichnung für den jeweiligen Circuit verwendet, um Mehrdeutigkeiten zu verhindern (Circuit-ID). Die folgenden 2 Bytes werden nicht verwendet und daher auf „00“ gesetzt. Auch die Flags im Kontrollbyte werden nicht verwendet und daher auf 0 gesetzt.

Im Info-Teil des Connect-Request-Frames wird zuerst die vorgeschlagene Fenstergröße für den Transport-Layer übertragen (1 Byte). Darauf folgt das Rufzeichen des Users, der den neuen Circuit aufzubauen wünscht. Schließlich folgt das Rufzeichen des Absenderknotens. Zusammen sind das also 15 Bytes im Info-Teil des Connect-Request-Frames.

Bei neueren Versionen von TheNet wird zusätzlich übertragen, bei welchem Netzknoten der User in das Netz eingestiegen ist, und über welchen Digipeaterweg er dort zu erreichen ist. Zuerst wird hierfür das Rufzeichen des Netzknotens übertragen, bei dem der User in das Netz eingestiegen ist. Danach folgt die eventuell vorhandene Digipeaterliste beim Uplink, also maximal 8 weitere Rufzeichen. Zum Abschluß folgt 1 Byte „00“. Für diese Erweiterung werden also minimal 8 und maximal 64 Bytes zusätzlich im Informationsteil übertragen.

Für alle beim Connect-Request übertragenen Rufzeichen wird das um 1 Bit nach links geschobene AX.25-Rufzeichen-Format verwendet.

### **Connect-Acknowledge.**

Auf ein Connect-Request-Frame wird mit einem Connect-Acknowledge-Frame geantwortet. Die ersten beiden Bytes des Transport-Kontrollblocks bezeichnen den Circuit des Knotens, der den Connect-Request gesendet hat. Sie werden direkt aus dem Connect-Request übernommen. Die darauf folgenden beiden Bytes bezeichnen den Circuit des Knotens, der den Connect-Acknowledge-Frame sendet, wieder in der Reihenfolge Circuit-Index - Circuit-ID, wie beim Connect-Request-Frame.

Wenn der Connect-Request akzeptiert wurde, sind alle Flags im Kontrollbyte gelöscht. Falls kein Circuit aufgebaut werden kann, ist das Choke-Flag (BIT 7) gesetzt.

Mit einem Connect-Request-Frame wird ein neuer Circuit angefordert. Die ersten beiden Bytes des Transport-Kontrollblocks spezifizieren den Circuit des Absenderknotens. Dabei gibt das erste Byte einen Index in der Circuit-Tabelle an (Circuit-Index). Das 2. Byte wird als zusätzliche Kennzeichnung für den jeweiligen Circuit verwendet, um Mehrdeutigkeiten zu verhindern (Circuit-ID). Die folgenden 2 Bytes werden nicht verwendet und daher auf „00“ gesetzt. Auch die Flags im Kontrollbyte werden nicht verwendet und daher auf 0 gesetzt. Im Info-Teil des Connect-Request-Frames wird zuerst die vorgeschlagene Fenstergröße für den Transport-Layer übertragen (1 Byte). Darauf folgt das Rufzeichen des Users, der den neuen Circuit aufzubauen wünscht. Schließlich folgt das Rufzeichen des Absenderknotens. Zusammen sind das also 15 Bytes im Info-Teil des Connect-Request-Frames.

Im Info-Teil des Connect-Acknowledge-Frames wird 1 Byte übertragen für die zu verwendende Fenstergröße im Layer 4.

### **Disconnect-Request.**

Wenn ein Information-Transfer-Frame empfangen wird und keine Information für die Gegenstation vorliegt, wird der Empfang mit einem Information-Acknowledge-Frame bestätigt. Die ersten beiden Bytes des Transport-Kontrollblocks bezeichnen wieder den Circuit des Knotens, an den der Information-Acknowledge-Frame gesendet wird, in der Reihenfolge Circuit-Index - Circuit-ID. Das dritte Byte des Information-Acknowledge-Frames ist nicht verwendet und wird daher auf „00“ gesetzt. Das 4. Byte des Transport-Kontrollblocks ist die Empfangs-Sequenz-Nummer und gibt die als nächstes von der Gegenstation erwartete Sendesequenznummer an.

Ein Disconnect-Request-Frame enthält keinen Info-Teil.

**Disconnect-Acknowledge.**

Als Antwort auf ein Disconnect-Request-Frame wird ein Disconnect-Acknowledge-Frame gesendet. Die ersten beiden Bytes bezeichnen wieder den Circuit des Knotens, an den der Disconnect-Acknowledge-Frame gesendet wird, in der Reihenfolge Circuit-Index - Circuit-ID. Die folgenden beiden Bytes werden nicht verwendet und deshalb auf „00“ gesetzt. Auch die Flags im Kontrollbyte werden nicht verwendet und daher gelöscht.

Ein Disconnect-Acknowledge-Frame enthält keinen Info-Teil.

**Information-Transfer.**

Ein Information-Transfer-Frame überträgt die Informationen des Users. Die ersten beiden Bytes bezeichnen den Circuit des Knotens, an den der Information-Transfer-Frame gesendet wird, in der Reihenfolge Circuit-Index - Circuit-ID. Darauf folgt ein Byte für die Sende-Sequenz-Nummer dieses Frames. Das 4. Byte des Transport-Kontrollblocks ist die Empfangs-Sequenz-Nummer und gibt die als nächstes von der Gegenstation erwartete Sendesequenznummer an. Damit werden gleichzeitig mit der Informationsübertragung die von der Gegenstation empfangenen Frames bestätigt.

Ist das More-Follows-Flag gesetzt, bedeutet dies, daß nicht die vollständige Information eines User-Frames auf einmal übertragen werden konnte, da im Information-Transfer-Frame höchstens 236 Bytes Info enthalten sein können (die restlichen 20 Bytes werden für Netzwerk - und Transport-Kontrollblock benötigt). Im nächsten Frame ist die Fortsetzung der User-Information enthalten.

Das NAK-Flag wird gesetzt, wenn ein empfangenes Information-Transfer-Frame nicht die erwartete Sendesequenznummer hatte. Damit wird die Wiederholung des fehlenden Frames angefordert.

Ist das Choke-Flag gesetzt, bedeutet dies, daß vorerst keine weitere Information für diesen Circuit angenommen werden kann.

Im Info-Teil des Information-Transfer-Frames wird die User-Information übertragen (max. 236 Bytes je Frame).

**Information-Acknowledge.**

Wenn ein Information-Transfer-Frame empfangen wird und keine Information für die Gegenstation vorliegt, wird der Empfang mit einem Information-Acknowledge-Frame bestätigt. Die ersten beiden Bytes des Transport-Kontrollblocks bezeichnen wieder den Circuit des Knotens, an den der Information-Acknowledge-Frame gesendet wird, in der Reihenfolge Circuit-Index - Circuit-ID. Das dritte Byte Des Information-Acknowledge-Frames ist nicht verwendet und wird daher auf „00“ gesetzt. Das 4. Byte des Transport-Kontrollblocks ist die Empfangs-Sequenz-Nummer und gibt die als nächstes von der Gegenstation erwartete Sendesequenznummer an.

Das NAK-Flag wird gesetzt, wenn ein empfangenes Information-Transfer-Frame nicht die erwartete Sendesequenznummer hatte. Damit wird die Wiederholung des fehlenden Frames angefordert.

Ist das Choke-Flag gesetzt, bedeutet dies, daß vorerst keine weitere Information für diesen Circuit angenommen werden kann. Das More-Follows-Flag wird nicht verwendet und ist daher gelöscht. Ein Information-Acknowledge-Frame enthält keinen Info-Teil.

DF6LN (23.7.93); erweitert um ON5ZS-Router ON5ZS/DG9FU (9.9.96)

**FlexNet-Interface:**

Allgemeines:

Die vorliegende Beschreibung dokumentiert die Implementation der FlexNet-Gateway-Routinen in TNN. Bei der Erarbeitung der Grundlagen bestand keinerlei Absichten, den FlexNet-Router zu kopieren, zu verbessern oder zu übernehmen. Ziel ist und war, eine engere Verknüpfung der TNN-Digipeater und FlexNet-Digipeater auf Systemebene zu erreichen und Network-Informationen gegenseitig auszutauschen.

Bis dato konnten die FlexNet - und Nord><Link Gruppen kein gemeinsames Interlink-Protokoll ausarbeiten. Aus diesem Grunde haben wir nun versucht, die Interlink-Kommunikation der FlexNet-Schnittstelle teilweise nachzubilden und damit eine vereinfachte Gateway-Schnittstelle zu realisieren.

Grenzen der Implementation:

Da die FlexNet-Schnittstelle nicht (öffentlich) dokumentiert ist, übernehmen wir keinerlei Funktionsgarantie. Aus den oben erwähnten Gründen bestand sogar die Absicht einer eingeschränkter Emulation. Bei Bedarf (und weitergehenden Informationen) kann die Schnittstelle zu einem späteren Zeitpunkt erweitert werden.

Realisiert wurden folgende Funktionen:

- Initialisierung,
- Laufzeitmessung zum Nachbarn,
- Beantwortung der Laufzeitmessung des Nachbarn,
- Auswertung der Destinations-Informationen,
- Rückmeldung bekannter Destinations,
- Find Destinations,.

Nicht realisiert sind folgende Funktionen:

- Header Kompression.

Diese Version kann nun beliebig viele FlexNet-Nachbar verwalten. Laufzeitmessung und Destinations-Übernahme erfolgen dynamisch. Die empfangenen FlexNet-Ziele werden in der gleichen Tabelle gespeichert wie die Nodes.



**ANHANG G:****Die GNU-Version von TNN****Einleitung:**

Unter MS-DOS hatten wir bisher immer damit zu kämpfen, daß der verfügbare Speicher auf 1 MB (+ HMA) beschränkt war. Diesen Speicher mußten sich alle Programme teilen. Damit waren Programme wie TNN in einen Speicherbereich eingezwängt.

Mit der GNU-Version steht nun der gesamte Speicher zur Verfügung. Aber warum noch MS-DOS unterstützen, wo es so gute Betriebssysteme wie OS/2, LINUX oder WIN 95 gibt ? Alle diese Betriebssysteme haben einen recht hohen Speicherbedarf, außerdem schleppt man meist eine für unsere Zwecke unnötige Oberfläche mit. Deshalb die weitere Unterstützung von MS-DOS.

Was bedeutet DPMI ?

Die ersten Prozessoren der Firma INTEL konnten 1 MB Speicher adressieren, dies war zu damaligen Zeiten eine Menge. Mit dem Boom der Computertechnik wurde schnell erkannt, daß diese Grenze schnell gesprengt würde. Deshalb implementierte INTEL eine CPU-Betriebsart, den Protected Mode. Mit dieser Methode konnte man auf einem 286'er (damals) unglaubliche 16 MB adressieren. Leider war man nicht recht konsequent mit der Umsetzung, so konnte man die CPU ohne weiteres in den PM (Protected Mode) schalten, aber zurückschalten in den Real Mode, so bezeichnet man den 8086 kompatiblen Modus, war nur mit großem Aufwand möglich. Erst mit dem 386'er wurde dieses Manko behoben. Außerdem wurde der Protected-Mode noch mal erweitert, so daß bis zu 4 GByte (32 Bit Selektor und 32 Bit Offset) adressieren kann.

Leider hatte dies keine Vorteile für ein Betriebssystem wie MS-DOS, durch die spezielle Art des PM würde es keine Abwärtskompatibilität geben. Um trotzdem auf den Speicher über 1 MB zugreifen zu können, wurden Krücken wie LIM-EMS oder XMS eingeführt. Hierbei wird ab dem 386'er ein weiterer Modus des 386'er ausgenutzt, der Virtuelle-8086-Modus, auf diesen möchte ich aber nicht weiter eingehen.

Um dieses Problem zu beheben, wurde das Dos-Protected-Mode-Interface definiert. Damit ist es möglich, Protected-Mode-Programme auch unter MS-DOS auszuführen, ohne gleich ein neues Betriebssystem implementieren zu müssen.

Der DPMI-Server schalten den Prozessor (ab 386'er) immer dann in den Real-Mode zurück, wenn z.B. eine DOS-Routine oder ein BIOS-Aufruf stattfinden. Danach wird die CPU wieder in den PM versetzt. Außerdem stellt der DPMI-Server eine gewisse Anzahl an Funktionen zur Verfügung, die das Programmieren im PM erleichtern.

Warum Protected ?

Adresse zu bilden.

Der Protected-Mode ist eigentlich für Multitasking-Systeme gedacht. Durch diesen Modus bietet die CPU Möglichkeiten, einzelne TASK voreinander zu schützen d.h. ein Programm kann nicht in ein anderes eingreifen. Dies ist aber nur durch einen gewissen Aufwand zu erreichen. So zeigt ein Pointer nicht auf eine physikalische Adresse, sondern enthält einen Selektor und einen Offset. Zum Generieren einer physikalischen Adresse im PM wird zum Selektorteil, der logischen Adresse, als Index in eine Deskriptortabelle gesprungen. Der Eintrag in die Deskriptortabelle enthält eine 24-Bit Basisadresse, welcher der Offset-Teil der logischen Adresse hinzuaddiert wird, um die physikalische Adresse zu bilden.

Im DPMI 32-bit ist dies ähnlich, allerdings werden dort 32-Bit-Offsets verwendet (4 GByte !), die einen Bezug zu einem 4G-langen-Segment herstellen können. Versucht jetzt eine Anwendung auf Daten jenseits einer Segmentgrenze zuzugreifen, so gibt es eine allgemeine Schutzverletzung (GBP oder General Protection fault), daher Protected Mode. In DPMI-Version von TNN wird der 16-Bit-DPMI verwendet.

Was ändert sich für den Nutzer?

Erstmal nichts, die DPMI-Version bietet erstmal die gleiche Funktionalität wie die „normale“ Version. Allerdings ist die Systemanforderung etwas höher. Durch das Umschalten zwischen PM und Real-Mode ist eine gewisse Geschwindigkeitseinbuße zu bemerken. Auch sollten min. 3-4 MB-RAM verfügbar sein. Für DOS-Aufrufe stehen dann allerdings mehr als 400 KByte zur Verfügung und die Buffer können etwas großzügiger bemessen werden. Auftretende Fehler (Exceptions) werden auf der Festplatte im TEXTCMD-Pfad verewigt und sollten an eine NORD><LINK-Service-Stelle (z.B. DG9FU,DB2OS oder DG1KWA) weitergeleitet werden, damit wir die Möglichkeit haben, den Fehler ausfindig zu machen.

Übrigens verträgt sich die DPMI-Version bestens mit OS/2 oder WIN 3.1x / WIN 95.

**GRENZEN / PROBLEME:**

Bedingt durch den Aufbau des DPMI gibt es einen ganz großen Haken. Wie oben bereits beschrieben muß die CPU fast laufend zwischen dem Protected Mode und dem Real Mode hin und herzuschalten. Dieses kostet natürlich Zeit.

Deshalb sind einige Sachen zu beachten:

**KISSLINK**

Nur mit Fifo, ohne hat es nicht viel Sinn !!

**BAYCOM-USCC**

Bedingt durch den Aufbau der Karte werden sehr viele Interrupts erzeugt ! Vier 9k6-Links entsprechen ca. einem 38k4 KISSLINK ohne Fifo ! Die USCC-Karte verfügt zwar auch über einen kleinen Fifo, ob der aber reicht ist fraglich !

**VANESSA**

Super für DPML, da die VANESSA keine Interrupts erzeugen !

**ETHERNET**

Macht natürlich IRQ ...wie kritisch dies ist, kann ich erst nach Tests sagen.....

Einen 486DX2-66 oder höher würde ich mal wärmstens empfehlen.

**DEBUG.TXT:**

Bei Abstürzen / Exceptions:

**Wird AUF DEM BILDSCHIRM** eine Debug-Ausgabe erzeugt. SOFORT NACH der TNN17x.EXE sollte das Programm „SYMIFY -O DEBUG.TXT TNN.EXE“ aufgerufen werden, dieses wertet den Stack aus und generiert die eigentliche Debug-Meldung, diese bitte an eine TNN-Service-Stelle weiterleiten. Der Sysop ist für einen Reset danach, selbst verantwortlich ! Dies kann entweder ein Hardware-Watchdog sein oder eine Datei in der START.BAT die nach dem Aufruf der TNN-EXE und SYMIFY steht, z.B. RESET.COM. Leider war eine andere Lösung bis jetzt nicht möglich !

Mögliche Exceptions:

08 = Doppelfehler,  
0A = Ungültiges TSS,  
0B = Segment nicht vorhanden,  
0C = Stack-Fehler,  
0D = Allgemeine Schutzverletzung,  
0E = Seitenfehler.

Weitere Hinweise zur 32bit-Version von TheNetNode für DOS:

- die TNN176.STA/MHEARD.TAB/CONVERS.PRS der 32bit-Version unterscheiden sich durch das DWORD-Alignment des GNU-Compilers zu den DOS/DPML-Versionen
- alle Funktionen der TNN171 für DOS/DPML sollten genauso bei der 32bit-Version funktionieren
- die 32bit-Version generiert keine Resets sondern verläßt das Programm, in einer Batch-Datei ist dann nach der TNN.EXE ein Programm aufzurufen, das den Reset durchführt. Zwischen TNN.EXE und RESET.EXE ist das „SYMIFY“ einzufügen (s.u.)

**Compiler:**

Als Compiler kommt der DJGPP/GNU zum Einsatz. Dieser Compiler ist frei verfügbar.

Weiter sind die folgenden Programme erforderlich:

**SYMIFY.EXE**

Zum Auslesen des Speichers bei einem Programmfehler.

**CWSDPML.EXE**

Der nötige DPML-Handler incl. CWSDPML.DOC, CWSDMP0.EXE CWSPARAM.EXE.

**RESET.EXE**

damit die Software auch sofort wieder gestartet wird.

Die START.BAT könnte nun so aussehen:

```
set tz=UTC0
set conversd=P:1
set tnn32buffers=9000
set tnnpath=c:\tnn\
set tnncfg=1009,32
set msgpath=c:\tnn\msg\
set confpath=c:\tnn\
set digicall=DB0EAM
set satqth=J041PI
set qth=9:16:12/51:21:10
set tokencom=0
set kiss1=1
c:\b-log.exe c:\tnn\sysop.pro -m

STARTCNT
IF ERRORLEVEL==1 GOTO OKGNU32
ECHO Starte alte TNN171...

REM !!!!!!!!!!!!!!! ALTE SOFT HIER STARTEN (TNN171.EXE) !!!!!!!!!!!!!!!
COPY TNN176.DOS TNN176.TNB
TNN_ALT.exe
REM !!!!!!!!!!!!!!! ALTE SOFT HIER STARTEN (TNN171.EXE) !!!!!!!!!!!!!!!

GOTO ENDE
:OKGNU32

ECHO Starte neue TNNGNU.EXE

REM !!!!!!!!!!!!!!! HIER WIRD DIE NEUE SOFTWARE GESTARTET !!!!!!!!!!!!!!!
COPY TNN176.DPI TNN176.TNB
TNNGNU.exe
SYMIFY -o DEBUG.ERR TNNGNU.EXE
TYPE DEBUG.ERR >> C:\TNN\TEXTCMD\DEBUG.TXT
REM !!!!!!!!!!!!!!! HIER WIRD DIE NEUE SOFTWARE GESTARTET !!!!!!!!!!!!!!!

:ENDE
RESET.EXE
```

#### **Weitere Fragen ?**

Dann sind zu empfehlen:

- PC Intern,
- Die 386/486'er-CPU,
- <http://www.delorie.com>.

ANHANG H:-----  
**D A M A - ein neues Verfahren für Packet-Radio !**  
-----

von DK4EG Detlef J. Schmidt,  
Stenbrechstr. 22,  
D-38106 Braunschweig

NORD><LINK e.V.  
c/o Peter Gülzow, DB2OS  
Allensteiner Str. 5  
D-30880 LAATZEN  
Germany

Immer wieder haben Packet-Radio-Amateure Probleme damit, daß sie vermeintlich nicht an 'ihren' Digipeater herankommen oder meinen, der Digipeater würde sie nicht verstehen. Und das, obwohl sie den Digi mit lautem gutem Signal aufnehmen können. Der Schluß, daß der Empfänger des Digi wohl taub sein müsse, scheint sich aufzudrängen.

Den Fall gibt es natürlich auch, er soll aber hier nicht betrachtet werden.

Wahrscheinlicher ist es, daß der Digipeater nicht 'nichts hört', sondern daß er 'viel zu viel hört'. Er hat ja typischerweise den 'besten' Standort aller beteiligten Stationen. Das führt dann zu Kollisionen beim Digi mit mehreren Stationen, die sich untereinander nicht hören ('Hidden-Stations-Problem'). Nur wenn eines der gleichzeitig empfangenen Signale deutlich stärker beim Digi ankommt als alle anderen, hat es Chancen 'durchzukommen'. Für weiter entfernt liegende Stationen ist es dann oft unmöglich, mit ihren Packets zum Digi durchzudringen.

Verschiedene Versuche sind schon unternommen worden, um dieses Problem auf den Amateurbändern zu lösen. Ein Lösungsansatz ist zum Beispiel der Einsatz von Duplex-Digis (BTMA). Aber auch dieses System hat etliche Nachteile. Der Hardwareaufwand eines Duplex-Digis ist um einiges höher als bei einem Simplex-Digi. Außerdem belegt er **z w e i** Kanäle, bringt aber nur den Durchsatz **e i n e s** kollisionsfreien Kanals. Eine Durchsatzerhöhung tritt nur statistisch durch Kollisionsreduzierung auf.

Elegant wäre es aber wohl, wenn man z.B. durch einfachen EPROM-Tausch oder noch besser nur durch eine andere Parametereinstellung das Problem lösen könnte.

Dieses Problem der Packet-Radio-Amateure ist nun aber gar nicht so neu. Es gibt auch andere Funkdienste, die das gleiche Problem haben, z.B., wenn von Schiffen auf hoher See ein Kommunikationssatellit angesprochen werden soll.

Ein etabliertes Verfahren, das dieses Kollisionsproblem löst, ist z.B. das DAMA-Verfahren. Es ist die Abkürzung von: Demand Assigned Multiple Ales, also etwa anforderungsbezogener Vielfachzugriff (auf den Übertragungskanal). Vereinzelt taucht in der Literatur auch der treffendere Begriff CODAC dafür auf (Carrier Oriented and Demand Assigned Controll). Der Begriff ist aber wohl wegen der phonetischen Verwechslungsmöglichkeit mit einem Firmennamen aufgegeben worden.

Der Ablauf dieses etablierten DAMA ist etwa folgender: In einem Connection oriented protokoll wird zunächst nach dem Slotted-Aloha-Verfahren versucht, den Satelliten zu connecten. Kollisionen sind selten, werden aber in dieser Phase in Kauf genommen. Danach kennt der Satellit die Station und nimmt sie in seine Polling-Liste auf. Es werden also alle Stationen reihum vom Satelliten aus aufgerufen (gepollt), wobei der Aufruf gleichzeitig die Bestätigung der empfangenen Packets ist.

Ist die Station einmal connected, sendet sie ihre Datenblöcke (I-Frames) nur noch nach Aufforderung durch einen Poll. Es können auch mehrere Frames in einem Block gesendet werden. Antwortet der Endknoten (User) nicht unmittelbar innerhalb einer vorgegebenen Zeit, wird angenommen, daß der Poll nicht angekommen ist, und im nächsten Durchlauf wird sofort der Poll wiederholt.

Desgleichen wird nach Empfang von Nutzdaten-Blöcken (I-Frames) in der nächsten Runde (also wenn alle anderen Stationen der Connect-Liste abgearbeitet sind) mit dem Poll die Bestätigung geschickt. Kommt dagegen vom Endknoten nur eine leere Bestätigung an (Receive Ready/Final), so wird er in der nächsten Runde übergangen.

Mit zunehmender Belegung des Übertragungskanals kann eine momentan nicht aktive Station noch weiter in der Poll-Priorität heruntersetzt werden, erlangt aber sofort wieder die höchste Priorität, wenn sie mal mit einem (oder mehreren) I-Frames antwortet.

Liest man sich die vorige Protokollbeschreibung durch, glaubt man fast, das AX.25-Level-2-Protokoll darin zu erkennen. Und darin liegt wohl auch die Chance und die Möglichkeit des DAMA-Verfahrens für Amateur-Packet-Radio. AX.25-L2 bietet unmittelbar alle Elemente, die für DAMA notwendig sind. Es müssen keine neuen Syntaxelemente eingeführt werden. Viele Funktionen lassen sich bereits durch Verstellen der Betriebsparameter erreichen. Der Rest ist z.T. nur eine minimale Änderung der Statetable in der Firmware.

Wie könnte nun eine DAMA-Version für Amateur-Packet-Radio aussehen ?

Da bereits alle Syntaxelemente vorhanden sind, müssen auch keine neuen eingeführt und erklärt werden. Wir bleiben daher für die Beschreibung in der bisherigen Terminologie von AX.25.

Die verschiedenen Phasen des Protokolls sollen hier einzeln betrachtet werden.

### **Verbindungs Aufbau**

Soll via Digi ein Endknoten connected werden, so wird die neue Station direkt in die Poll-Liste aufgenommen und mit SABM-Frames angepollt. Erfolgt innerhalb einer vorgegebenen Anzahl von SABM kein UA des angesprochenen Endknotens, wird die Station wieder aus der Poll-Liste gestrichen. Geht die Initiative umgekehrt vom Endknoten aus, wird wie bisher im CSMA-Verfahren ein SABM an den Netzknoten gesendet. In dieser Phase sind Kollisionen möglich, es ist also eventuell notwendig, nach einem vorgegebenen Timeout wie bisher weitere SABM zu senden, bis der Netzknoten mit einem UA antwortet.

Damit ist der Endknoten in die Polling-Liste (Userliste bei TheNet) aufgenommen und wird ab diesem Zeitpunkt koordiniert. Der Endknoten antwortet mit einem RR count 0 als Signal an den Digi, daß sein UA aufgenommen wurde.

### **Idle**

Solange kein Informationstransfer zwischen Netz- und Endknoten stattfindet (Ideln), sendet der Netzknoten seine Polls mit RR und dem zugehörigen aktuellen Count. Kommt als Antwort des Endknotens beim Digi ein RR mit dem zugehörigen Count an, dann wird die Zeit bis zum nächsten Poll an denselben Endknoten verlängert, um den Kanal nicht unnötig mit Polls zu belasten. Wird das RR des Endknotens nicht beim Digi empfangen (weil entweder der Poll oder die Antwort verloren ging), dann wird derselbe Endknoten beim nächsten Durchlauf (also wenn alle anderen Uplink-Stationen einmal abgearbeitet wurden) sofort wieder angepollt usw.

Wenn nach einer vorgegebenen Anzahl von Polls keine Antwort empfangen wird, kann der Endknoten aus der Poll-Liste gelöscht werden. Dieses Verfahren stellt nur auf den ersten Blick mehr Transferaufwand dar als in der bisherigen Version, da auch nun schon die Gegenstation routinemäßig angepollt und irgendwann abgehängt wird. Der Unterschied ist nur, daß in der neuen Version der Endknoten früher aus der Userliste gestrichen wird, wenn er nicht mehr antwortet, bei gleichbleibender Anzahl von Versuchen.

### **Nutzdaten-Transfer Netzknoten-Endknoten**

Dieser Fall unterscheidet sich in nichts vom 'normalen' CSMA-Verfahren. Da die Initiative ohnehin beim Digi liegt (Master), kann er auch zu jeder Zeit statt eines Polls ein oder mehrere I-Frames an den Endknoten (Slave) senden. Der Endknoten bestätigt wie gehabt mit RR und dem zugehörigen Count, er kann aber auch direkt mit I-Frames und den korrespondierenden Counts bestätigen. Auch die Anwendung des Poll-/Final-Bits bleibt unverändert.

### **Nutzdaten-Transfer Endknoten-Netzknoten**

Wie schon zuvor skizziert, sendet der Netzknoten reihum Polls an alle in der Userliste verzeichnete Endknoten. Der Endknoten wartet also, bis er vom Digi mit einem RR angepollt wird oder ihm ein I-Frame gesendet wird und antwortet dann unmittelbar mit einem oder mehreren I-Frames zum Netzknoten. Dieses Warten auf den Poll macht den entscheidenden Aspekt der Kollisionsverhütung aus im Gegensatz zum bisherigen Verfahren, bei dem mehrere Endknoten quasi gleichzeitig senden können, weil sie sich untereinander nicht hören und nicht koordiniert sind.

Zusätzlich wird das Problem behoben, daß zwei Endknoten senden, die sich eigentlich hören könnten, aber in der jeweils eigenen Totzeit zwischen dem Abschalten des Empfängers und dem Abstrahlen des modulierten Trägers die andere Station nicht 'hören'. Dieser Fall ist gar nicht so selten, da verschiedene sendebereite Stationen ja von dem Träger des Digi untereinander synchronisiert werden.

Der Digi wird die gesendeten I-Frames nicht sofort bestätigen, sondern erst in der nächsten Runde, nachdem alle anderen Stationen der Liste bedient wurden. Dies ist dann auch gleichzeitig der nächste Poll. Stehen keine I-Frames zum Senden an, wird vom Endknoten wieder nur ein RR mit Count geschickt.

## **Verbindungsabbau**

Es kommen wieder dieselben Syntaxelemente zum Einsatz. Will der Netzknoten (Digi) den Endknoten (User) abwerfen, so schickt er einfach wie bisher ein DISC-Frame. Der Endknoten antwortet darauf unmittelbar mit einem UA (Finalbit gesetzt). geht das UA verloren, antwortet der Endknoten beim nächsten eintreffenden DISC-Frame wie bisher mit einem DM-Frame. Will im umgekehrten Fall der Endknoten 'aussteigen', so wartet er den nächsten Poll ab und sendet daraufhin sein DISC-Frame an den Netzknoten. In diesem Falle ist es unerheblich, ob der Digi sofort oder erst in der nächsten Runde mit UA bestätigt.

## **UI-Frames**

Eine gewisse Sonderstellung nehmen UI-Frames ein (Unnumbered Information), sowohl im normalen CSMA-Verfahren als auch im DAMA-CSMA-Verfahren. Sie sind ursprünglich einmal dafür kreiert worden, einen Informationsaustausch außerhalb des normalen Protokollablaufes zu realisieren.

Werden UI-Frames vom Endknoten gesendet, so kann dies normalerweise nur außerhalb des Betriebs mit dem Digi auftreten. Eigentlich sollte es diese Fälle ja nicht geben, da in diesem Falle der Direkt-QSO nicht die Digi-Einstiegsfrequenz benutzt werden sollte. UI-Frames des Digi stellen ohnehin wieder kein Problem dar, da er ja von allen Stationen auf der Frequenz gehört wird.

## **Sonstige Protokollelemente**

Somit wäre eine idealisierte Session vom Verbindungsaufbau bis zum -abbau komplett beschrieben.

Es wurden nicht alle AX.25 möglichen Protokollelemente dargestellt. Dies ist aber in diesem Kontext auch nicht zwingend notwendig, da sämtliche Elemente ihre logische Bedeutung behalten. DM, RNR, REJ werden in dem gleichen Sinne benutzt wie bisher. Der Unterschied zur reinen CSMA-Version ist lediglich, daß der Endknoten sie wie die anderen Protokollelemente nur nach einem Poll des Digi sendet.

Netzknotenseitig gibt es nur den Unterschied, daß diese entsprechenden Frames nicht sofort oder zu beliebiger Zeit, sondern immer erst dann gesendet werden, wenn alle anderen registrierten Stationen (im gleichen Sinne) einmal abgearbeitet wurden. und genau der im Frame adressierte Endknoten angesprochen werden soll.

## **Verträglichkeit der Protokollversionen**

Eine mögliche Einführung des DAMA-CSMA-Verfahrens könnte sich kontinuierlich vollziehen. Natürlich wird die Durchsatzsteigerung um so größer, je mehr Stationen auf das neue Verfahren einsteigen. Aber selbst Stationen, die nicht auf die neue Kanal-Zugriffsmethode einsteigen, können teilweise zur Durchsatzsteigerung beitragen, indem sie ihre Parameter anders einstellen. Dazu müßte der Delay zwischen Empfang eines an sie gerichteten Frames und der zugehörigen Bestätigung (T2 oder DWAIT) auf einen Minimalwert gesetzt werden

(weniger als 1 sec).

Die Zeit bis zur Pollaussendung auf ein eigenes ausgesendetes I-Frame (um eine Bestätigung 'anzumahnen') wird auf einen relativ hohen Wert gesetzt, der deutlich größer ist als normalerweise der zeitliche Abstand zwischen zwei Polls des Netzknotens.

Um die vollen Möglichkeiten des Verfahrens auszuschöpfen, müssen beide Seiten die neue Protokollversion fahren. Dazu muß der Endknotenseite irgendwie mitgeteilt werden, daß sie sich nun auf die neue Protokollvariante einzustellen hat. Mehrere Methoden bieten sich an:

- Einführung eines UPLINK-Kommandos zusätzlich zum CONNECT-Kommando;
- Einführung eines kanalspezifischen Parameters im TNC, der die Protokollvariante steuert;
- automatische Erkennung der Protokollvariante mittels Protokoll-Identifizier-Bytes vom Netzknoten;
- eingeschränkte Nutzung des Verfahrens durch Änderung der TNC-Timer (wie zuvor skizziert);
- Daneben ist natürlich auch noch die Einführung eines neuen Protokollelements möglich, mit der die jeweilige Variante gesteuert wird. Das würde z.B. wie bei X.25 die Funktion eines SABM-Frames darstellen. Auch damit wäre weiterhin Kompatibilität zu dem bisherigen Protokoll gewährleistet, da die 'alten' Protokollversionen auf unbekannte Frames nicht oder mit FRMR reagieren sollen

**Zusammenfassung**

Das skizzierte Verfahren kann den Durchsatz auf einem AX.25-Kanal deutlich erhöhen. Es hat die entscheidenden Vorteile, daß es kein Netzzusammenbruch durch Überlastung des Kanals gibt. Die Übertragungskapazität des Kanals steigt kontinuierlich bis zum Maximum, es gibt keinen Umkipppunkt wie bei reinem CSMA, bei dem die Netto-Übertragungsrate oberhalb einer gewissen Schwelle (ca. 65 bis 85 % bei CSMA pur) wieder abnimmt.

Nur bei geringer Kanalbelastung werden bei DAMA mehr Übertragungen notwendig sein (Overhead). Dies stört aber nicht weiter, solange die Kanalkapazität nicht ausgeschöpft wird. Mit zunehmender Kanalauslastung geht der Overhead-Anteil dann weiter zurück bis auf den minimalen Wert des idealen CSMA-Verfahrens ohne Kollisionen.

Als entscheidende 'soziale' Komponente gibt DAMA auch entfernteren Stationen die Möglichkeit, stabil über den Digi zu arbeiten, ohne von den näherliegenden Stationen 'niedergebrüllt' zu werden.

Es ist nicht notwendig, daß alle beteiligten Stationen gleichzeitig auf die neue Protokollvariante umsteigen. Alle Protokollelemente behalten ihre ursprüngliche Bedeutung, so daß die Endknoten die alte und die neue Variante nebeneinander benutzen können. Der Kanaldurchsatz wird dabei um so besser, je mehr Stationen anteilig die neue Variante benutzen.

**Literatur**

- div. : Normblatt X.25 CCITT.
- div. : Transaction on communication IEEE.
- Fox, T. : AX.25 level 2 protocol specification AMRAD.
- Kauffels, F.J.: Lokale Netze . R.-Müller-Verlag.
- Schmidt, D.J. : Synchrone DFÜ-Protokolle mit 6809-Micro-Computern in heterogenen Sternnetzen. TU-Script BS'81.
- Tanenbaum, A. : Computer Networks. Prentice Hall Verlag.
- Schmidt, D.J. : DAMA, ein neues Verfahren für Packet-Radio cq-DL, 4/89

**Glossar**

- DM : Disconnect Mode
- DISC : Disconnect Frame
- FRMR : Frame Reject
- I : Information Frame
- REJ : Reject Frame
- RNR : Receive Not Ready
- RR : Receive Ready
- SABM : Set Asynchronous Balanced Mode
- SARM : Set Asynchronous Response Mode
- UA : Unnumbered Acknowledge
- UI : Unnumbered Information Frame
- BTMA : Busy Tone Multiple Access
- CSMA : Carrier Sense Multiple Access

**Connection oriented protocoll:**

Alle Knoten der Strecke kennen alle Stationen, welche die Strecke benutzen, zumindest kurzzeitig. Im Gegensatz dazu steht das Connectionless protocoll, bei dem alle Daten-Packets einfach nur weitergereicht werden (also z.B. einfaches Level-2-Digipeaten bei PR). Connection oriented Protokolle sind typischerweise aufwendiger in der Realisierung und benötigen mehr Rechnerleistung in den Netzknoten, dafür sind sie aber speziell auf Übertragungskanälen mit geringer Kapazität überlegen, da nicht in jedem Frame die gesamte Routing-Information mittransportiert werden muß, sondern nur einmal während des Verbindungsaufbaus.

**Definition des DAMA-Slaves nach DG2FEF**

Im folgenden hier in Stichpunkten die Definition des DAMA-Slaves, wie er zur Zeit in TF2.7 implementiert ist. Diese Definition soll in Zukunft verbindlich für die Programmierer von Layer-2-Software sein, um einen einheitlichen Standard zu schaffen, auf dessen Basis die Betriebsarten DAMA und OPTIMA (DAMA für Duplex-Digis) weiterentwickelt werden können.

- 1.) Als Master-Poll (bzw. DAMA-Poll) ist jedes an den Slave adressierte Frame zu verstehen, in dessen Adressfeld zugleich das DAMA-Bit, das Poll/Finalbit und das Command/Response-Bit gesetzt ist.
- 2.) Eine Station soll in den Betriebszustand „DAMA-Slave“ wechseln, wenn sie ein an sich adressiertes Frame mit gesetztem DAMA-Bit empfängt. Als DAMA-Slave soll sie einen globalen Timer führen, der mit einem geeigneten Wert (der vorgeschlagene Mindestwert beträgt 4 Minuten) zu initialisieren ist. Der Timer soll mit jedem an sie gerichteten Frame mit gesetztem DAMA-Bit neu auf den Startwert initialisiert werden. Die Station soll den Betriebszustand „DAMA-Slave“ verlassen, wenn entweder der globale DAMA-Timeout ausläuft, oder sie alle Verbindungen auf dem Kanal des DAMA-Masters aufgelöst hat.

- 3.) Der Slave darf für ein QSO mit dem oder über den Master niemals ein Frame ausstrahlen, bei dem zugleich das Poll/Final- und das Command/Response-Bit gesetzt sind. Das gilt auch für sogenannte „Unproto-Polls“.
- 4.) Es darf keine vom Slave motivierten Aktionen geben. Jede Aktion des Slave darf nur als Reaktion auf den Master-Poll und die in ihm übermittelten Informationen erfolgen. Eine Reaktion auf den T3 ist zulässig, in diesem Zusammenhang auch die Aussendung eines Polls, jedoch sollte der T3 grundsätzlich auf eine Zeit nicht kleiner als 3 Minuten initialisiert sein. Auch sollte dieser Parameter nicht für den User konfigurierbar sein.  
  
Eine Ausnahme bildet das Auf- oder Abbauen eines neuen Links, hier findet natürlich eine vom Slave motivierte Aktion statt. Der T1 darf in diesem Fall als Kriterium für die Erzeugung eines SABM oder DISC benutzt werden, die Aussendung muß innerhalb der Reaktion auf einen Master-Poll erfolgen.
- 5.) Die Reaktion des Slave auf einen Master-Poll hat unverzüglich und ohne Beachtung der DCD zu erfolgen.
- 6.) Die Reaktion des Slave auf einen Master-Poll muß aus einem einzigen, zusammenhängenden Sendedurchgang bestehen. Ein „Flackern“ des Senders darf unter keinen Umständen auftreten.
- 7.) Bei Multi-Connect-Verbindungen, sendet der Slave bei einem DAMA-Poll auf einem beliebigen Link alle anstehenden, ungesendeten Frames aller bestehenden Links aus. Schon mal gesendete Frames eines bestimmten Links werden dagegen nur dann wiederholt, wenn dieser Link explizit angepollt wird.



ANHANG I:**TCP/IP und TNN**

Eine Einführung zur Einstellung der Routen ab der Version TheNetNode V 1.73

Es hat sich gezeigt, daß viele Sysop von TNN Dingis bisher nichts oder nur wenig mit TCP/IP zu tun hatten. Aus diesem Grund versuche ich hier eine kleine Einführung mit Beispielen zu schreiben.

Es sollen keine Grundlagen des TCP/IP Protokolls vermittelt werden, sondern es soll nur dazu dienen einen TNN Digi TCP/IP mäßig richtig zu konfigurieren. Die TCP/IP Grundlagen sind an anderer Stelle nachzulesen.

IP Adresse:

Zuerst muß der Digi eine IP Adresse bekommen.

IP Adressen werden weltweit international vergeben, d.h. eine IP Adresse ist nur EINMAL auf der Welt vorhanden bzw. zugeteilt. Man kann sich also nicht einfach eine Adresse aussuchen, sondern muß sich an den entsprechenden IP Adreßverwalter wenden.

Weltweit macht das Brian Kantor (Rufzeichen gerade nicht zur Hand). Weiterhin gibt es in jedem Land einen weiteren IP Adreßverwalter. In DL ist das im Moment DH9KAE, der wiederum weitere Adreßverwalter hat, welche die einzelnen Regionen verwalten.

Für den Hamburger Raum macht das z.B. Eddi dk3uz. Übrigens hat Brian Kantor das gesamte 44er Netz und muß dafür bezahlen! Ich weiß nicht wieviel, aber solche Netze sind gebührenpflichtig; das nur kurz zur Info.

Es gibt eine Ausnahme:

44.128.0.0 - 44.128.255.255 sind Adressen die jeder mal zum testen verwenden darf, denn diese Adressen werden offiziell nicht zugeteilt.

Wie sieht so eine IP Adresse aus ?

Beispiel von db0hht.ampr.org:

```

44.130.0.100
|   |   |   +--- die 100 ist die Adresse von db0hht.ampr.org
|   |   +----- die  0 steht für das Netz 0
|   +----- die 130 steht für DL
+----- die  44 steht für die Domain ampr.org, also Amateurfunk

```

Noch ein weiteres Beispiel von db0nds:

```

44.130.150.100
|   |   |   +--- die 100 ist die Adresse von db0nds.ampr.org
|   |   +----- die 150 steht für das Netz 150
|   +----- die 130 für DL
+----- die  44 steht für die Domain ampr.org, also Amateurfunk

```

In diesem Beispiel unterscheiden sich die Adressen an einem Punkt, d.h. einmal steht da die 0 für Netz 0 und einmal die 150 für Netz 150!

Die IP Adresse wird bei TNN jetzt mit dem Kommando IPA <Adresse> eingestellt.

Einstellen der tcp/ip routen:

Wie erreichen wir jetzt, daß IP Pakete von z.B. db0nds aus in das Netz 0 richtig geroutet werden ?

Für das Netz 0 ist db0hht.ampr.org das Gateway, d.h. alle Pakete sollten über db0hht.ampr.org geschickt werden und zwar die Adressen 44.130.0.0 bis 44.130.0.255, also die 255 Adressen für das entsprechende Netz. DB0HHT.ampr.org reicht diese Adressen dann richtig auf dem Einstieg weiter. Wir müssen also ein sog. Clusterrouting einstellen und das können wir z.B. bei db0nds so machen:

Zuerst wird die Route zu db0hht.ampr.org eingestellt (44.130.0.100)

IPR 44.130.0.100 + NET/ROM

Dann werden alle Adressen des Netzes 0 über db0hht.ampr.org geroutet:

IPR 44.130.0.0/24 + NET/ROM 44.130.0.100

Jetzt können alle Rechner im Netz 0 von db0nds aus über db0hht.ampr.org erreicht werden. Wenn da nicht noch etwas wäre.

Woher weiß db0nds denn wie db0hht mit ax25 erreicht werden kann?

Dafür ist der ARP Eintrag vorgesehen.

ARP 44.130.0.100 + NET/ROM db0hht

Damit wird der Software mitgeteilt, daß TCP/IP Pakete, welche die Adresse 44.130.0.100 haben, an die AX25 Adresse db0hht per NetRom zu schicken sind!

Wir müssen hier nämlich genau zwischen „Hostname“ und „AX25-Adresse“ unterscheiden! Das ist ein ganz wichtiger Punkt, der gerne Verwirrung stiftet:

Es kann z.B. folgendes sein:

Hostname : dk3hg.ampr.org

AX25-Call: dk3hg

Username : dk3hg

Hier taucht immer dieser „dk3hg“ auf, aber alle drei Namen, die zwar gleich lauten, haben unterschiedliche Bedeutung!!

Noch ein Wort zu den „routen“:

Es macht wenig Sinn einzelne Hosts zu routen, ohne das Clusterrouting anzuwenden, denn dann erreicht man nur diesen EINEN Hostnamen und weiter nichts. Das ist aber nicht der Sinn eines Routers, wie ihn TNN173 darstellt.

Außerdem sollten die Sysop sich untereinander absprechen „wer wie was“ routet, damit es nicht zum Durcheinander kommt. Ein guter Anfang ist eigentlich immer sich an die Sysop der eigentlichen TCP/IP Digi zu wenden, wenn es um solche Fragen geht.

Einen ganz besonderen Vorteil möchte ich noch herausstellen. Bisher war es vor allem für den Norden (Netromland) sehr problematisch TCP/IP in den Süden (FlexNet-Land) zu machen, weil die Digi im FlexNet-Land immer ihren Broadcast über FlexNet nach Norden schicken mußten. Das kann man jetzt umgehen, wenn die TNN, die an FlexNet angeschlossen sind, ihre routen richtig eingestellt haben. Dann ist es möglich von z.B. db0hht aus auch alle möglichen anderen TCP/IP Digi zu erreichen, die NUR FlexNet machen.

Ich möchte mich auf diesem Wege auch bei Frank dg3aah bedanken, der mit viel Geduld bei db0nds die routen für uns eingestellt hat.

Wer wissen möchte welcher Subnetzkoordinator für seine Region zuständig ist, kann in den Boxen nach „Subnet\_Koordinator\_Liste“ suchen. Dort sind alle Koordinatoren für DL aufgeführt. Ich schicke diese Liste auch gerne auf Anforderung per Mail zu, sollte sie nicht mehr in den Boxen zu finden sein.

Wer in seiner Region keinen Subnetzkoordinator hat, muß sich dann an DH9KAE wenden, um ein neues Subnetz zu beantragen. Gleichzeitig sollte ein OM in dieser neuen Region die Verwaltung dieses Netzes übernehmen. Es empfiehlt sich dann eine IP Adresse für den TNN Router zu nehmen, die mit einer 1 anfängt, wie z.B.

44.130.neues\_Netz.1. Das hat dann den Vorteil das man dieses Netz bei Bedarf noch weiter aufteilen kann.

Eine aktuelle Liste der IP Adressen ist bei db0hht oder bei anderen UNIX Digi die WAMPES verwenden, im Verzeichnis /tcp zu finden. Je nach Sysop mehr oder weniger aktuell :-). Wer die gesamte weltweite Liste haben möchte und einen Internetanschluß hat, kann sich bei ftp://ftp.ucsd.edu/hamradio mal umschauen.

Im Anhang 1 ist die Aufteilung eines Netzes in weitere Subnetze zu finden. Im Anhang 2 ist ein unvollständiges Beispiel wie wir es bei db0nds eingestellt haben.

Anhang 1

Man kann auch ein Netz von z.B. 44.130.0.0 bis 44.130.0.255 noch weiter in einzelne Subnetze aufteilen. Das sieht dann z.B. so aus:

Netmask	/Bits	Adressen	Hosts
44.130.0.0/24		256	254
44.130.0.128/25		128	126
44.130.0.192/26		64	62
44.130.0.224/27		32	30
44.130.0.240/28		16	14
44.130.0.248/29		8	6
44.130.0.252/30		4	2

## Anhang 2

ipa 44.130.150.100

ipr 44.130.0.100 + netrom  
arp 44.130.0.100 + db0hht  
ipr 44.130.0.0/24 + netrom 44.130.0.100

ipr 44.130.82.101 + Ruhner-B.  
arp 44.130.82.101 + Ruhner-B. VC db0oca-11 db0lwl-2  
ipr 44.130.83.0/24 + Ruhner-B. 44.130.82.101

ipr 44.130.20.0/24 + Ruhner-B. 44.130.82.101  
ipr 44.130.42.0/24 + Ruhner-B. 44.130.82.101  
ipr 44.130.56.0/24 + Ruhner-B. 44.130.82.101  
ipr 44.130.60.0/24 + Ruhner-B. 44.130.82.101  
ipr 44.130.80.0/24 + Ruhner-B. 44.130.82.101  
ipr 44.130.88.0/24 + Ruhner-B. 44.130.82.101  
ipr 44.130.91.0/24 + Ruhner-B. 44.130.82.101

Wird noch erweitert ....., soll nur als Anregung dienen :

Bei Fragen, Kommentaren, Verbesserungsvorschläge etc. bitte ich um Mail an mich.  
Michael (dk3hg@db0hht)

## IP-Router

Diese Anleitung richtet sich an Sysop und User, die etwas über IP, IP-Routing und IP-Nutzung mit TheNetNode erfahren möchten. Alle Beschreibungen sind auf den Kern reduziert und genügen sicher nicht erfahrenen Gurus.

### 1. WAS IST IP?

IP ist ein bestimmtes Frameformat, das Datenaustausch über ein Medium ermöglicht. So ein Medium kann z.B. ETHERNET (Netzkarte), ARCNET (dito) oder aber auch PACKET RADIO sein. Dies ist ein wichtiger Aspekt. Sowohl AX.25 als auch NET/ROM sind für IP nur Transportmedien. Deshalb ist TCP/IP auch nicht zum Open Systems Interconnection (OSI) Normvorschlag konform.

DIENSTE	TELNET, FTP, HTTP
TRANSPORT	TCP, UDP
ROUTER	IP
MEDIUM	ETHERNET, ARCNET, SLIP, PACKET-RADIO

TheNetNode kennt nur ROUTER und MEDIUM (und noch ROUTER-Dienste, aber das ist hier erstmal egal).

### 2. IP-ROUTER

Ein IP-Router analysiert die Zieladresse von IP-Frames und versucht anhand dieser Adresse, Frames zuzustellen. Hierzu benutzt TNN die sogenannte IP-Router-Tabelle, die mit dem Befehl IPR konfiguriert wird.

IPR Adresse/Maske [+/-] [Flags] [Interface] [Gateway]

Adresse/Maske definiert eine Adresse (Maske=32, also alle Bits gelten) oder einen Adreßbereich, für den dieser IPR-Eintrag gilt.

Für Flags kann D wie „dynamisch“ angegeben werden. Diese Adressen haben eine besondere Bedeutung, die unten näher erklärt wird.

Als Interface kann ein Portname oder „NETROM“ angegeben werden. Frames für diese Adresse(n) werden IMMER an dieses Interface gesendet.

Gateway wird im Zusammenhang mit ARP erklärt. Es ist die IP-Adresse des nächsten Routers, wenn diese Adresse nicht direkt erreicht werden kann oder soll.

### 3. ARP

ARP bedeutet ADDRESS RESOLUTION PROTOCOL. Wenn ein Frame anhand der IP-Routen-Tabelle einem Interface(=Port) zugeordnet wurde, wissen wir immer noch nicht die AX.25-Adresse (=Rufzeichen) der Station, die dieses Frame bekommen soll, wir brauchen also eine Zuordnung IP-Adresse und Rufzeichen.

ARP DestIP [+/-] [Publ.] [Port] [DG/VC] [Call] [Digi1[Digi2]]

DestIP ist die Adresse (hier wird nur EINE Adresse angegeben, eine AX.25-Adresse hat auch nur eine IP Nummer). Ist DestIP ein Gateway, dann wird die Adresse hier angegeben und in den IP-Routen immer auf diese Adresse als Gateway verwiesen. Dadurch können Adreßbereiche einzelnen Gateways zum routen zugeordnet werden.

Für Publ. kann ein „P“ angegeben werden, dies bedeutet, das dieser Eintrag öffentlich ist, und Anfragen von anderen Stationen über diesen Eintrag beantwortet werden.

DG/VC bestimmt, ob die Verbindung Connected oder mit UI-Frames passieren soll.

Call/Digi gibt den Connectweg an.

Für NET/ROM wird für Port wieder „NETROM“ angegeben.

### 4. Automatisches ARP

Das Eintragen der ARP-Daten kann automatisch erfolgen, wenn das Ziel direkt vom Knoten gehört wird, also alle lokalen User usw. Es geht nicht für via-Verbindungen und NET/ROM.

Wenn TNN ein Frame auf einem Port zustellen will, und keinen passenden ARP-Eintrag hat, wird eine UI-Anfrage generiert. Wird diese beantwortet, dann werden künftige Frames an diese Adresse geschickt. Der Eintrag gilt für 60 Minuten.

### **5. Dynamische Adressen**

Ein User kann mit „IPR +“ für die aktuelle Verbindung eine IP-Adresse beantragen, wenn für diesen Port dynamische Adressen eingetragen sind. Die Adresse ist für eine Stunde vergeben (ab der letzten Nutzung).

Ein Skript zum IP-Betrieb könnte also so aussehen:

IPR +

[IP-Adresse wird als Text ausgegeben, diese lesen]

ARP +

[Verbindung ist umgeschaltet]

[LOKAL auf IP schalten und loslegen]

DB7KG

**LIZENZ:****Allgemeine Lizenz für Amateurfunk Software (ALAS)**

Copyright(C) 1992 Nord><Link e.V.  
Andreas Wißkirchen  
Berrenrather Str.537  
50937 Köln

Dieses Dokument darf beliebig vervielfältigt oder verteilt werden, solange es nicht verändert wird. Für Anregungen, wie es zu verbessern ist, bin ich dankbar.

1. Vorwort:

Diese Lizenz entstand aus der General Public Licence der Free Software Foundation (GPL). Ich habe versucht, den Sinn zu erhalten und mehr Klarheit hineinzubringen. Einige Passagen sind vollständig entfallen. Es ist aber jedem Nutzer freigestellt, an Stelle dieser Lizenz die GPL Bedingungen anzuwenden. Der Sinn dieser Lizenz ist es, den Autor und den Anwender der Software zu schützen. Es sind daher einige Einschränkungen erforderlich, und es entstehen auch einige Pflichten für den, der die mit dieser Lizenz verbundene Software weitergibt oder verändert. Dies wird dadurch erreicht, daß die Software durch Copyright geschützt wird und der Nutzer durch diese Lizenz die Möglichkeit einer nahezu unbeschränkten Nutzung erhält.

2. Geltungsbereich

Diese Lizenz gilt für jedes Programm oder Teil eines Programmpaketes, das eine Copyright-Notiz ausgibt, die sich auf diese Lizenz bezieht. Im folgenden bedeutet „Programm“ entweder das Programm oder einen Teil davon. „Du“ bist der Lizenznehmer.

3. Deine Rechte:

Du darfst das Programm nutzen oder kopieren oder verteilen oder verändern, solange Du damit keine kommerziellen Absichten verbindest.

4. Deine Pflichten:

4.1. Du darfst den Copyrightvermerk und den Hinweis auf diese Lizenz nicht verändern, und er muß bei jedem Start des Programms eindeutig für den Benutzer sichtbar sein.

4.2. Du mußt jedem Dritten, dem Du eine Kopie des Programms gibst, die gleichen Rechte einräumen, die auch Dir gegeben wurden. Du mußt ihm auch die gleichen Pflichten auferlegen.

4.3. Du darfst für die Weitergabe kein Geld verlangen außer den Kosten für das Medium und Porto.

4.4. Du darfst das Programm nur komplett weitergeben, so wie Du es bekommen hast.

4.5. Wenn Du das Programm veränderst oder Teile davon für eigene Arbeiten verwendest - wörtlich oder verändert - so gelten die folgenden Punkte für das daraus entstehende neue Programm:

4.5.1.

Du mußt deutlich sichtbar angeben:  
- Deinen Namen und Deine Adresse und  
- die Tatsache, daß Du das Programm geändert hast oder Teile des Programms verwendet hast.

4.5.2.

Du mußt das Programm entweder mit dem kompletten Quelltext weitergeben oder jedem auf Verlangen eine Kopie des Quelltextes gegen eine Gebühr von maximal der Kosten des Mediums und der Portokosten aushändigen.

4.5.3.

Du darfst keine Beschränkungen aussprechen, die über diese Lizenz hinausgehen.

5. Sonstiges

Du erhältst das Programm ohne jede Garantie für Funktion, Fehlerfreiheit oder Anwendbarkeit für eine bestimmte Sache. Du verzichtest auf jede Schadenersatzforderung, gleich aus welchem Grunde. Mit der Nutzung des Programms erkennst Du diese Lizenzbedingungen vorbehaltlos an.

Vers. 1.0, 13-OCT-92

**Ende ALAS****ABSCHLUSS:**

Was noch genauer erklärt werden muß, werden die (hoffentlichen) Fragen zeigen. Dieses File, so weiß ich schon jetzt, wird noch öfters geändert und ergänzt werden müssen.

Mit freundlichen Grüßen

Nord><Link e. V.  
Andreas Wißkirchen  
Berrenrather Str.537  
50937 Köln

Ausdrucke (vom Laserdrucker) gibt es gegen Unkostenerstattung von DM 0,08/Seite + Porto bei:

Henning Happe  
Sechsäcker 20  
34246 Vellmar.

Geführt von	Henning	DG9FU.
Beiträge von	Peter	DB2OS
	Andreas	DB7KG
	Georg	DF2AU
	Nils	DF6LN
	Klaus	DF7BZ
	Andreas	DG1KWA
	Frank	DG3AAH
	Bernd	DG8BR
	Odo	DL1XAO
	Kurt	DL2LAY
	Rainer	DL5HCD
	Matthias	DL9HCJ
	Peter	HB9PAE
	Daniel	ON5ZS.

Mein besonderer Dank geht an DF6QQ. Uli hat in mühsamer Kleinarbeit sich der Interpunktion angenommen und auch noch etliche kleine Schreibfehler beseitigt!

Diese Ausgabe wurde Korrektur gelesen von:

Andreas	DG1KWA.
Nils	DF6LN