

NORD><LINK

User Documentation for TheNet Network Software  
Version 1.0 dated 05.27.88

Originators : DB2OS Peter, DC4OX Michael, DF2AU Georg.  
Edited by : DB2OR Peter  
Reprints with CC allowed for noncommercial use  
-----

1. Commands for the User  
-----

1.1 USER  
-----

After entering 'U' (for USER) you should see:

```
H:DB0FD> TheNet 1.0 (736)
Uplink(DB2OS) <--> Circuit(BS:DB0FC-8 DB2OS)
Uplink(DC4OX) <..> Downlink(DC4OX-15 DC8AM)
Circuit(BS:DB0FC DF2AU) <--> Circuit(DJ9KD-10 DF2AU)
Uplink(DB2OR) <..> CQ(DB2OR-15)
Circuit(BS:DK4EG-1) <--> Host(H:DB0FD)
Uplink(DF3AV)
```

DB0FD is the official node-id of the TheNet-node.  
H is the "Ident" of the node, a kind of pseudonym.

This for once shall be a geographical information which allows to figure out the area where a unknown call is coming from. The Ident in addition to the function of the call-sign allows multiple connecting of nodes. So you can connect H, H-1, H-2, and so on at the same time but only DB0FD and not DB0FD-1, DB0FD-2. The other way, to do multiconnect on a TheNet-node, is to use your own call-sign with different SSIDs at the same time - but that is not supported by any TNC firmware nowadays.

We also use the car license plate id of the area and for a 70cm-node, "70" is added. That seems most meaningful to us because other alternatives like ZIP-codes are not very mnemonic. This is NOT an abuse of call-signs because idents don't have a format following international standards. That is the same a a call to CQ or a test or a 4-digit call with Amtor which are not considered wrong used of the call-sign. The required use of call-sign every 10 minutes is taken care for and the idents in the AX.25 address field are limited to end-users and in between the nodes, the official call-sign is used. Every TheNet node can also be identified using the INFO, USER and NODES commands. The use of area license plate ids would increase the transparency of our networks considerably.

736

is the actual number of free buffers in the node. One buffer consists of 32 bytes data and a 4 byte list-pointer, 36 bytes in total. Every stored information used up one buffer and the list of other known nodes takes up buffers too. The reason why this # of free buffers is made known to the user, is to tell him not to make any attempts if the # of buffers available is very small. Because, if TheNet does not have any free buffers, a reset is activated which clears all pending connects causing total loss of stored information. During regular operation, this case is almost impossible, as with TheNet parameters set to the default, there will never be a shortage of buffers (you cannot load many packets after another into the node).

### Uplink

indicates that the user with call-sign in () entered the network on this node.  
An eventually used digipeater is not shown.

### Downlink

indicates that the user with call-sign in () exits the network on this node.  
An eventually used digipeater is not shown.

### Circuit

on the left side means that the user with call-sign in () is coming from the following node and has entered that node with the call-sign as follows.

### Circuit

on the left side means that a connection to the node mentioned in () before is active and that the user there has the following call-sign.

### Host

means a connection to the operator terminal of the node.

### CQ

can only show on the right side and indicates that user "CQ" is calling.  
To activate a connection the call on the right hand side has to be issued in a connect command.

<-->

indicates an active connection.

<..>

indicates a connection being established.

An entry without a right hand portion means that this is where the connection ends right now and the user is connected with the command interpreter of this node.

## 1.2 NODES

-----

If you enter 'N' (for NODES) you should see:

H:DB0FD> Nodes:

H70:DB0FD-7                    DB0FC                    DB0FE                    DL0TEL                    DB0JD

The shown idents:call-signs are other TheNet-nodes which can be reached by entering C H70 or C DB0DJ e.g. The path taken to such a node is of no interest; it is known to H (DB0FD) and that is enough. In order to get to DB0DJ you don't have to enter C DL0TEL and from there C DB0DJ as DB0DJ can only be reached via DL0TEL.

To learn about the exact path to DB0DJ you can enter:

N DB0DJ, and you should see:

```
H:DB0FD> Routes to: DB0DJ
  0 2 0 DL0TEL via DJ4KW
```

That means that the path to DB0DJ from DB0FD leads to DL0TEL first via the non-TheNet-Digi DJ4KW. To learn about the rest of the path you have to connect to DL0TEL, entering C DL0TEL. The fact that you get there via an ordinary Digi is known to H and doesn't have to be entered.

One also had to enter DJ4KW in addition, if DL0TEL is not known via DK4KW so its not showing up in NODES commands. At DL0TEL you can enter N DB0DJ again and so trace the rest of the path.

Once again: This method is only necessary if you really have to know the path or if node DB0DJ is unknown to H. In order to run a QSO via node DB0DJ known to H, DB0DJ is sufficient.

Now there is one question:

"If the path to DB0DJ is known to DB0FD, why can't I be told that when entering N DB0DJ ?"

That makes sense, but TheNet is organized in a way, so that a path is known only so far that the node knows about the destination and the next node to reach that destination.

If all nodes are organized that way, the path is completely defined.

This organization is meaningful because storing the complete path with many nodes involved used up too much space and the information is not really needed to build the connection.

Say you want to establish a connection to DB0DJ and enter C DB0DJ. The result might be:

```
H:DB0FD> Failure with DB0DJ
```

You could have guessed that in advance knowing about the parameters in detail:

```
H:DB0FD> Routes to: DB0DJ
  0 2 0 DL0TEL via DJ4KW
  | | |
  | | +----- "Port"- Indicates if the path is via the packet-channel on to (0) or
  | |           via RS232 on the node directly to the next node (1, e.g. H-> H70).
  | |
  | +----- "obsolescence count" - for the user meaningful in so far 0 indicates
  |           that the path is fixed, i.e. won't be affected by automatic updates
  |           in TheNet.
  |
  +----- "quality" - Quality of the connection. This is the most important
  parameter for the user. 255 is max. and means it's a 99% connection,
  128 stands for 50%, a real bad connection which certainly won't
  connect. (0 for newly established connections may also mean that a
  quality index is not available yet - and there is not much chance of
  a connection either).
```

If you enter N DL0CCC, you might see:

```
H:DB0FD> Routes to: DL0CCC
> 0 2 0 DL0CCC via DJ4KW
```

The '>' preceding Route tells you that this Route is in use. Which doesn't mean a lot in this example as the quality was too bad for a connect. The '>' only means a lot if there are several paths (Routes) to a given destination and then it indicates which one is in use. TheNet automatically tries the one with the highest quality first.

### 1.3 INFO -----

If you enter I (for INFO) you should see:

```
BS:DB0FC>
NORD><LINK
Braunschweig <JO52FG>
144.625 MHz, 5W, GP
OPs: DF2AU, DK4EG 9 DK0MAV
```

This shows the identification of a node on request. So it is always possible to find out which node is involved and what it's official node-id is if it was connected with the ident pseudonym. For a better orientation in the network, the ident usually is preset by the caller using the license plate id of the area. In addition, TheNet automatically sends a beacon to "ID", to comply with the law.

### 1.4 ROUTES -----

If you enter R (for ROUTES) you should see:

```
BS:DB0FC> Routes:
  0 DB0FD 0 0 |
  0 DB0FE 0 0 |
> 1 DB0FC-7 248 1
> 1 DB0FC-7 248 15
```

This command gives an overview which paths are in use from this Digi right now and what their load is.

for every entry in the list of neighbors, it shows:

- \* ">" in case there is an active connection
- \* the used port (0=HDLC Port, 1=RS232 Port)
- \* the path to this neighbor (Call + eventually digipeater)
- \* the quality of the path (255 = best, 0 = worst)
- \* the # of connections which exist to this neighbor
- \* "|" if the entry is fixed

If you only want to see the path to a specific neighbor, you have to enter:

```
ROUTES 1 DB0FC-7
BS:DB0FC> Routes:
> 1 DB0FC-7 248 27
```

## 1.5 PARMS

-----

If you enter P (for PARMS) you'll see:

```
BS:DB0FC> 50 1 100 248 6 5 1800 15 300 2 12 180 8 8 900 64 10 5 2 10 100
18000 0 1 2 1
```

For users with an IBM PC and the program (TURBO PACKET) from DL1BHO the PARMS are evaluated right away and that looks like:

```
BS:DB0FC> 50 1 100 248 6 5 1800 15 300 2 12 180 8 8 900 64 10 5 2 10 100
18000 0 1 2 1
```

NET/ROM PARMS-evaluation:

|                     |                   |                    |
|---------------------|-------------------|--------------------|
| 01: Max-Nodes 50    | 02: min-Quality 1 | 03: HF-Quality 100 |
| 04: V24-Quality 248 | 05: Obs-Init 6    | 06: min-BCast 5    |
| 07: Broadcast 1800  | 08: Lifetime 15   | 09: T-Timeout 300  |
| 10: T-Retry 2       | 11: T-AckDelay 12 | 12: T-BsyDelay 180 |
| 13: T-Window 8      | 14: NoAckBuf 8    | 15: Timeout 900    |
| 16: Persistence 64  | 17: SlotTime 10   | 18: FRACK 5        |
| 19: MAXFRAME 2      | 20: L2-Retry 10   | 21: T2-Timer 100   |
| 22: T3-Timer 18000  | 23: L2-Digi 0     | 24: CallCheck 1    |
| 25: ID-Beacon 2     | 26: CQ-UI-Frame 1 |                    |

This listing shows only constant values which only the operator of a system can change. There are no parameters which change during operation. The parameters listed above have turned out to be optimal. Otherwise it should be obvious that calling for those parameters all the time doesn't make much sense.

---

Par 01. Max. # of entries in the destination node list. Every entry of a dest. node takes up 36 Byte in storage, so this parameter takes care for limiting this list so there is enough frame-buffer-space left.

| default | min | max |
|---------|-----|-----|
| 50      | 1   | 400 |

---

Par 02. Worst quality for automatic update of the destination node list. 0 = no auto-update. The destination node list is automatically kept up to date. To include only meaningful paths within a certain quality range, only paths with the defined quality (this parameter) are used. If this parameter is 0, automatic update is completely switched off.

| default | min | max |
|---------|-----|-----|
| 1       | 0   | 255 |

---

Par 03. Channel 0 (HDLC packet-channel) quality. Quality of the packet-channel normally the channel, to which the radio is connected.

| default | min | max |
|---------|-----|-----|
| 100     | 0   | 255 |

---

Par 04. Channel 1 (RS232, direct crosslink) quality. If 2 or more nodes are attached directly via the RS232-interface, this is the quality of the RS232-channel.

| default | min | max |
|---------|-----|-----|
| 255     | 0   | 255 |

---

Par 05. Initial value of the aging-count, 0 = aging count is not maintained. If a new node is found or updated or if a successful connect to a destination node is done, the aging count for this node is set to its initial value. Any time the auto-update-round-robin-interval (see below) expires, without having had a connection with the destination node in between, this count is decremented.

| default | min | max |
|---------|-----|-----|
| 6       | 0   | 255 |

---

Par 06. Min. value of the count, for the path information to be sent to other nodes too. Only the nodes with a count exceeding this parameter are shared with neighbor nodes during automatic update. This parameter prevents the sharing of old paths. It should never exceed parameter 5, otherwise only the local node is shared with others.

| default | min | max |
|---------|-----|-----|
| 5       | 1   | 255 |

---

Par 07. Destination-list-auto-update round-robin-interval in seconds, 0 = switched off. Within this interval, the reachable destination nodes are shared and the counts for all nodes reached are decremented. If this parm is 0, no sharing of reachable nodes with others is done.

| default | min | max   |
|---------|-----|-------|
| 1800    | 0   | 65535 |

---

Par 08. Initial count of network packet-lifetime-counter. All network-(layer-4) packets have a lifetime count, which indicates how long this packet is supposed to live. It defines, how many times this packet may be handed over from node to node. With every transfer through a node, the count is decremented. If it gets to 0, the packet is thrown away. If a packet originates locally, the lifetime-counter is initialized with this parameter. The counter is to prevent a packet from being transferred throughout the network in an endless loop. It should be set slightly higher than the number of nodes on the longest known path.

| default | min | max |
|---------|-----|-----|
| 10      | 0   | 255 |

---

Par 09. Transport FRACK in seconds. Similar to FRACK on layer 2, but the time is much longer as the transport-layer starts at layer 2, so that no packets should get lost and retries should only be caused by brief interruptions (e.g. reset) or total loss of a node.

| default | min | max |
|---------|-----|-----|
| 300     | 5   | 600 |

---

Par 10. Transport RETRY. Similar to RETRY on layer 2. After this number of retries, it is assumed that a node is down. Because, as mentioned in 9., the

transport-layer starts at layer 2, this counter can only be exceeded if a node fails for a short time or completely.

| default | min | max |
|---------|-----|-----|
| 10      | 2   | 127 |

---

Par 11. Transport acknowledge delay in seconds. This number of seconds is used to wait for an acknowledge of incoming transport-layer packets. Its meaning is to eventually package the acknowledge into transport-layer packets to be sent thus saving one additional packet if one waits a while until maybe another transport-layer packet is to be sent.

| default | min | max |
|---------|-----|-----|
| 6       | 1   | 60  |

---

Par 12. Transport busy-delay in seconds. If there are many transport-layer packets arriving at a destination node (see 14.), the transport-layer goes busy or "choke". This state is communicated to the local node, so the node won't send any more packets. If the choke is released that will be communicated too. If the release info does not come through because maybe a node in the path is down, the flag for the choke state is deleted after the busy-delay time anyway, even if there was no release message.

| default | min | max  |
|---------|-----|------|
| 180     | 1   | 1000 |

---

Par 13. Transport window-size in # of frames. Number of the transport-layer packets not within the expected transport-send-sequencenumbers which are stored on receiving. The number of transport packets which can be sent in sequence without an acknowledge.

| default | min | max |
|---------|-----|-----|
| 4       | 1   | 127 |

---

Par 14. Overfill-limit in # of frames. Number of packets which can be stored on the transport-layer, until a choke-info is sent to the connected node. This identifies also the number of frames which can be stored on the link-layer, before the link-layer goes busy. This limit prevents an overload of a TheNet-node in case of too many incoming transport-layer packets or in case a station in a link is trying too many packets all at once.

| default | min | max |
|---------|-----|-----|
| 4       | 1   | 127 |

---

Par 15. No-activity-timeout in seconds, (0 = switched off). If there is no exchange of information on layer-2-link for that amount of time, the link is cancelled.

| default | min | max   |
|---------|-----|-------|
| 900     | 0   | 65535 |

---

Par 16. P-persistence value. This parameter serves for decision making for the transmitter control and to avoid collisions. Every time, a packet is going to be transmitted, you must wait until the channel is free (no DCD). If the channel is free, a random number between 0 and 255 is generated. If this random number is smaller or equal the p-persistence value, the TNC will acknowledge the PTT-line and data transfer is started. If the random number is not within the p-persistence range, the TNC won't start

transmitting and will wait for some time and then repeat the above algorithm. The duration of the delay is defined by the next parameter.

| default | min | max |
|---------|-----|-----|
| 64      | 0   | 255 |

---

Par 17. Slot time interval in 10ms. This parameter defines the duration of the time slot for the p-persistence-control. Every time the TNC transmits a packet and the random number mentioned in 16. is not within the des p-persistence-range, a wait is performed for the slot time and then the procedure is started all over.

| default | min | max |
|---------|-----|-----|
| 10      | 0   | 127 |

---

Par 18. User-Link FRACK in seconds. Defines the number of seconds between repetition of not acknowledged packets or between polls. If ordinary digipeaters are used, this number is calculated as follows:

$(\# \text{ of digipeaters} * 2 + 1) * \text{FRACK}.$

To avoid collisions (see FRACK-text in this paragraph) there is another multiplication with a random number  $\leq 1$ .

| default | min | max |
|---------|-----|-----|
| 5       | 1   | 15  |

---

Par 19. User-Link MAXFRAME in # of frames. Number of packets on layer-2 which may be sent in sequence without receiving an acknowledge.

| default | min | max |
|---------|-----|-----|
| 2       | 1   | 7   |

---

Par 20. User-Link RETRY, 0 fore always. Defines the number of retries to establish layer-2 contact to another station (answer on commands and poll). After this number of retries the link is recorded as down. 0 set on an 24h-QRV-station would cause a disaster, so be warned.

| default | min | max |
|---------|-----|-----|
| 10      | 0   | 127 |

---

Par 21. User-link T2 in milliseconds \* 10. This timer defines the waiting time following which an incoming packet is acknowledged with a RR/REJ/RNR-packet. This delay should improve throughput because there is a chance to send for someone also during that time and there is also a chance for the send-layer, to add a pending acknowledge into another packet thus saving one link-layer packet.

| default | min | max   |
|---------|-----|-------|
| 100     | 0   | 65535 |

---

Par 22. User-link T3 (timer 3 ) in milliseconds \* 10. This parameter defines the time after which layer-2 will check if a link is still active, if ther have been no activities. It is used similarly to the transport-layer to overcome deadlocks on layer-2.

| default | min | max   |
|---------|-----|-------|
| 18000   | 0   | 65535 |

---

Par 23. AX.25 digipeating on (1) / off (0). If switched to on, the node can be used as a "normal" digipeater. As normal digipeating is not used in TheNet and

has disadvantages over real TheNet-connections (see FRACK in another text in this section) it's not necessary. It only made sense at a time when there were almost only mailboxes which were doing store-and-forward operations following the fixed WORLI-scheme.

In the meantime there is mailboxsoftware from DF3AV (Die BOX) as well as WORLI which can handle TheNet too.

| default | min | max |
|---------|-----|-----|
| 0       | 0   | 1   |

---

Par 24. Call-sign-check on (1) / off (0). If set to on, the call-signs except the idents have to meet the following requirements when they are entered:

- a) The length of a call-signs must be from 4 to 6 characters.
- b) All characters must be numeric or alphanumeric.
- c) The call-sign must contain one or two numerics.
- d) The last character of the call-sign must be alphanumeric.
- e) SSID, if used, must be between 0 and 15.

| default | min | max |
|---------|-----|-----|
| 1       | 0   | 1   |

---

Par 25. Stations identification bake.

- 0 = NONE , no ID-Bake.
- 1 = AFTER , only after transmitting.
- 2 = EVERY , always.

I set to on, a packet by which the node identities itself with call-sign and ident is sent every 10 minutes. The necessity of this bake results from the use of an ident and also from the requirement for a special TheNet watchdog which has to become active after there has been no PTT activity for more than 10 minutes. Attention: If the parameter is set to "1" the ID-bake is transmitted only if there is any activity via the node or if the nodes-bake is transmitted. If the TNC does not transmit for some time, then there is no nodes-bake either. So if you are operating in watchdog mode, this parameter has to be set to "2".

| default | min | max |
|---------|-----|-----|
| 2       | 0   | 2   |

---

26. CQ-call UI-frame on (1) / off (0). If set to on, the optional text following the CQ-command is sent as UI-packet (with sender call-sign "call-15" and destination call-sign "CQ"). This does only make sense on the network entrance node.

| default | min | max |
|---------|-----|-----|
| 1       | 0   | 1   |

---

## 1.6 CQ

-----

By entering CQ you can start a CQ-call via every TheNet node.

SYNTAX: CQ  
CQ text... (optional with Text, not more than 75 char)  
(no additional digipeater allowed)

How do I start a CQ-call ?

Let's assume DB2OS in Hannover would like to place a general CQ-call to a node in Braunschweig. First he makes requests a connection to BS:DB0FC and there he enters the CQ-command:

example:

```
* c db0fd
* (1) CONNECTED to DB0FD *
```

```
c bs
H:DB0FD> Connected to BS:DB0FC
```

```
cq CQDX de DB2OS HANNOVER JO42VG/EM60G VIA BS -- PSE CONNECT DB2OS-15
```

IMPORTANT: By a command following it or a RETURN, the CQ-state is cancelled!

-----  
VERSION A)

-----  
OM Karl, DK7AL is connected to BS at the same time and will see the following data:

```
BS:DB0FC> TheNet 1.0 (731)
Uplink(DF3AV)                <--> Circuit(BS77:DB0FC-8 DF3AV)
Uplink(DF2AU)                <..> Downlink(DF2AU-15 DK4EG-1)
CircuitH:DB0FD DB2OS)       <..> CQ(DB2OS-15)
Uplink(DK7AL)
```

"<..> CQ(DB2OS-15)" indicates that DB2OS (via node H:DB0FD) is looking for a connection into the BS area and has entered the CQ-command.

DK7AL now has to enter only "C DB2OS-15" and will at once be connected to DB2OS !!! The troublesome backtracking of the path is not necessary any more.

-----  
VERSION B)

-----  
OM Wolfgang, DB3AN is monitoring the frequency and suddenly sees the following packets on the screen (WA8DED Firmware):

```
fm DB2OS-15 to CQ ctl UI^
CQDX de DB2OS HANNOVER JO42VG/EM60G via BS -- PSE CONNECT DB2OS-15
|           |
|           |
|           +----- CQ CQ CQ...probably with text
|
+----- sender, OM Peter, DB2OS
```

This packet has been transmitted from BS:DB0FC right after receiving the CQ-command (with text) as a UI-packet.

In order to answer the CQ-call, Wolfgang did not have to connect the node BS first, but enters with his TNC the command to establish a connection with DB2OS-15 (Connect DB2OS-15, careful with SSID |), as if DB2OS-15 was a direct neighbor. After BS:DB0FC received the SABM-packet from DB3AN, the connection with DB0OS is installed at once and on the other end there is the message:

"BS:DB0DC> Connected to DB3AN".

As you see. the connection can be build on the user-level of a node or directly after viewing the CQ-call.

## 1.7 CONNECT

-----

The connect command is used to establish a connection with another node, another user or with the host-terminal of the TNC. That's the sequence in which the command is analysed.

CONNECT DB0FD

means to scan the list of known destinations for the call DB0FD. If found, a connection is established and the following message is displayed:

BS:DB0FC> connected to DB0FD

If DB0FD cannot be found in the destination list, the node assumes that DB0FD must be another user and transmits SABMs to this call-sign.

If there is an acknowledge, the following message is displayed:

BS:DB0FC> connected to DB0FD

A user can also be connected via digipeater. Then you would have to enter:

CONNECT DB3AN via DC40X

The term "via" maybe omitted or abbreviated. Up to 8 digipeaters are possible. The call-signs then have to be separated by blanks.

To connect to a host-terminal you have to enter:

CONNECT

without further parameters

Potential error messages are:

Failure with .....: the destination requested did not answer

Busy from.....: the destination refused to establish a connection

The following messages indicate that the node is overloaded and that's why the connection could not be established:

Node busy

Link table full

Circuit table full

Host table full

A connect command can be cancelled any time by entering arbitrary data, e.g. a blank line.

## 2. Commands for the User

-----

### 2.1 NODES

-----

Syntax: NODES nodecall + ident quality Count Port neighbor (digicall...)  
or: NODES nodecall - ident quality Count Port neighbor (digicall...)

where:

nodecall = Destination call  
ident = Destination pseudonym (or \*, if unknown or none given)  
quality = path quality  
Count = Initial lifetime of this destination. If 0, the destination is assumed to be permanent.  
Port = Port to the next neighbor. 0 is the HDLC port, 1 is the RS232 port.  
neighbor = neighbor's call-sign, via which the destination shall be reached. If the destination cannot be reached directly, the call-sign of the destination has to be repeated.  
digicall = max. 2 digipeater for the path to the neighbor. The call-signs are separated by blanks.

The Nodes command allows for manually changing entries in the destination list or to delete them without defining new ones.

The command with "+" adds a new destination to the list or updates a given entry with new parameters.

The command with "-" deletes an entry. A destination will only vanish completely from the list after the last path to it has been deleted.

Invalid entries do not result in error messages but they are simple skipped.

## 2.2 INFO

-----

Syntax: INFO text

where: text = char. string, <CR> = 0D HEX as end-character.

The text may be max. 80 characters. Longer strings are cut off. The minimum length is 1 character. Deleting is only possible by entering another message. After a cold start, the message is empty. Thus a loss of power can be detected easily. The message is appended to the message contained in the EPROM (a user's I command).

## 2.3 ROUTES

-----

Syntax: ROUTES port nodecall (digicall...) + quality  
or: ROUTES port nodecall (digicall...) - quality

where:

port = port to the next neighbor. 0 is the HDLC port, 1 is the RS232 port.  
nodecall = Destination call.  
digicall = max. 2 digipeater for path to reach the neighbor. The call-signs are separated by blanks.  
quality = path quality.

Entries into the neighbors list are created automatically if the round robin call by the neighbor is received. They can also be created manually using a NODES + command. Every new entry is marked "free" and is set to the standard

quality of the used channel. With the ROUTES command, it is possible to update the quality and to lock the paths.

The command "+" locks the entry given by port, nodecall and digicall and sets the quality parm of this path.

If the referenced entry does not yet exist, it is created, initialized with the values mentioned above and locked.

The command "-" releases the entry. If the path to the destination is not used, the entry is deleted at once. Otherwise the parameters are used like in the "+" version. A not locked entry is deleted the very moment it is not used by a channel any more.

The quality of a path is used when searching for the best path to a given destination. By changing this entry using the ROUTES+ command, the search via certain neighbors can be made easier or more difficult.

By setting the quality of a path to 0 and locking it, you can force the system to ignore the path completely and even round robin calls by those neighbors won't then be received.

## 2.4 PARMS

-----

Syntax: PARMS parmlist

where: parmlist = sequence of new settings for the parameters in the same format as used with the PARMS command.

Entering "\*" instead of a value, won't change the parameter defined by this position.

The variables must be separated by blanks.

Using the PARMS command, the user can adjust the parameters of a node to actual requirements or do some experiments without having to program an EPROM for every change.

Invalid (i.e. not within the allowed range) values will cause a command abort. It is not necessary to enter the complete list all the time. It is okay to enter the values up to the last parameter to be changed.

As there are no error messages for invalid entries, the setting should be checked afterwards with the PARM command (without argument).

During all experiments, you should carefully consider in advance, what you want to achieve. With a bad combination of settings it is perfectly possible to block the whole node, even for users being actually logged on.

## 2.5 SYSOP

-----

Syntax: SYSOP

With the SYSOP command, a user request being accepted as a participating user. In return, he is sent 5 numbers from the node and has to answer giving the 5 numbers of his own password. There is no feedback about success or failure of this command.

It is perfectly legal to intentionally do several attempts with wrong passwords, followed by a correct one and then wrong ones again. That is to confuse others

which might be listening. The acceptance as a user will be kept until a disconnect.

## 2.6 HIGH

-----

Syntax: HIGH portnumber

where: portnumber = number of the port desired. The connect exit is 0,  
the status exit 1.

The exit portnumber will be activated. (relay on, LED on).

After a reset, both ports are inactive.

Using this and the LOW command, a power or antennae switch can be done.

## 2.7 LOW

-----

Syntax: LOW portnumber

where: portnumber = number of the port desired. The connect exit is 0,  
the status exit 1.

The exit port will be inactivated (relay off, LED off).

## 2.8 RESET

-----

Syntax: RESET

RESET causes a cold start. All RAM will be initialized. All lists are deleted, all connections (even the user's own) are disconnected. All parameters are initialized with the EPROM settings. The INFO is cleared. The password is fetched from the EPROM.

## 3. Commands which can only be entered from a Terminal

-----

All commands are started with a special character (see patch list). The default is <ESC> = 1B HEX. On access from a terminal, the user authorization is set automatically.

### 3.1. F

-----

The network works full duplex. This can be switched from the host terminal only. The command is <ESC> F 0 to switch to half duplex and <ESC> F 1 to switch to full duplex. Via a constant in the EPROM, you can select if the sender is quiet during breaks (PTT = off) or if flags are sent then.

### 3.2. C

-----

The terminal is connected to the command processor of the node. All commands like during a normal HF connect are possible.

### 3.3. D

-----

The terminal is disconnected from the command processor of the node.

### 3.4. P

-----

Syntax: P text

where: text = char. string. End char is <CR> = 0D HEX.

Using this command, you can enter a new password. It consists of max. 80 characters. All characters except <CR> = 0D and <LF> = 0A HEX are valid, even control characters. Upper and lower case makes a difference. Please consider in advance, if you will later be able to send those characters from your TNC! On SYSOP the blanks within a password are not checked.

The command P without argument, will show the current password framed by two asterisks.

### 3.5. T

-----

Syntax: T value

where: value = number from 0 to 255

T determines the delay from PTT active to the start of transmitting the first data via the HDLC port. The delay is (value \* 10 milliseconds).

The command T without argument displays the actual setting.

The default after reset is taken from the EPROM (300 ms):

### 3.6. Y

-----

Syntax: Y flag

where: flag = 0 or 1

The command Y allows external connect requests to be handled or inhibited ( the requestor gets a "Busy from ...").

Flag 0 = inhibits connections, flag = 1 permits them.

The command Y without argument displays the actual setting.

## 4. Remarks

-----

### 4.1 Default Settings

-----

All default parameter are contained in a list at the beginning of the EPROM. These also contain the call-sign, the sign of the digipeater and a default password. Thus a remote access of the owner is possible even after a complete loss of the operation parameters.

#### 4.2 Quality of a connection

---

The quality can be in the range of 0 to 255 and reflects the reliability and the speed of a channel, that is finally the overall throughput. The quality value is more or less a matter of experience.

quality / 256 \* 100% = quality in %

| Recommended values:                                | quality | % "perfect" |
|--|---------|-------------|
| 9600 Baud RS232 cable between two nodes            | 255     | 99%         |
| 9600 Baud RS232 diode matrix between 3 nodes       | 248     | 96%         |
| 9600 Baud Packet-channel, only 2 nodes have access | 240     | 93%         |
| 1200 Baud Packet-channel, only 2 nodes have access | 224     | 87%         |
| 1200 Baud Packet-channel, User and several nodes   | 192     | 75%         |
| 300 Baud Packet-channel, shortwave                 | 128     | 50%         |

rough estimate: 255 = 99% = "almost every packet makes it"  
128 = 50% = "only every second packet makes it"

The quality of a path via several nodes is composed of the single qualities of the connections:

Overall quality =  $(Q_1/256 * Q_2/256 * \dots * Q_n/256) / * 256$

Example : node 1 <-> node 2 = 255 (99%)  
node 2 <-> node 3 = 255 (99%)  
node 3 <-> node 4 = 224 (88%)  
node 4 <-> node 5 = 224 (88%)

gesamt  $(255/256 * 255/256 * 224/256 * 224/256) / 256 = 192 (75\%)$

---

Hints to make changes to a TNC for operation as a network node following DL1BHO

Clock rate of the SIO-SyncB-clock  
-----

If the TNC is operated with 2.4576 MHz, the BR1 connection on the board must be removed and SyncB (SIO Pin 29) must be connected to Pin 1 of 74HC4060. This change is also described in the TNC2c-guide.

If the TNC is operated with 4.9152 MHz, the following changes are necessary:

- Change quartz
- remove 27pF capacitor
- replace 100pF capacitor with 50pF
- Remove BR1 connection and SyncB (SIO Pin 29) to Pin 2 (Pin 2 !!!) of the HC4060. Pin 2 is not positioned close to BR1, but the change is necessary for the SyncB-clock to be 600 Hz.

Change for Operation with Network-Software  
-----

From Pin 2 of the MAX232, connect a 4,7K resistor to pin 8 of the MAX232 (often already in place in newer TNCs, ... beneath the board).

The connection of SIO-pin 22 has to be changed as follows:

- Remove the connection from SIO-pin 22 to the through contact
- 10K resistor from SIO-pin 22 to the through contact
- Connect SIO-pin 22 with a piece of wire to pin 23 of the V24-connector
- With a wire from pin 8 of the V24-connector to pin 10, tie pin 10 of the connector to +5V.

If only one connector cable between the TNCs is used, as it is recommended in the TheNet manual, then the wire from pin 10 to pin 23 ties pin 22 to +5V and the software is able to understand that a TNC is connected to the port.

Connector Cable for 2 TNCs Interlink Operation

| TNC 1        | TNC 2    | signal        |
|--------------|----------|---------------|
| +-----+      | +-----+  |               |
| 1  -----  1  | 1        | prot. ground  |
| 2  -----  2  | 2        | Tx data       |
| 3  -----  3  | 3        | Rx data       |
| 5  ----  5   | ---  5   | CTS           |
| 20  ----  20 | ---  20  | DTR           |
| 7  -----  7  | 7        | signal ground |
| 10  ----  10 | .---  10 | V-            |
| 23  ----  23 | .---  23 | DRS           |
| +-----+      | +-----+  |               |

