

Setting up KA-Node/Netrom network access

Jim Oberhofer KN6PE

Updated: April 2007

KA-Node/NetRom support was added as part of Outpost v2.1. This Application Node is an update to the original note and clarifies how to set up Outpost to take advantage of this feature.

Introduction

The following are excerpts from the TAPR website on network schemes, originally published in "Packet Radio: What? Why? How? / Articles and Information on General Packet Radio Topics," TAPR, Publication #95-1. 1995, by Greg Jones, WD5IVD (http://www.tapr.org/pr_intro.html#NetSchemes).

"The first networking scheme with packet radio was Digipeaters. Digipeaters would simply look at a packet, and if its call was in the digipeater field, would resend the packet. Digipeaters allow the extension of range of a transmitter by retransmitting any packets addressed to the digipeater.

"Kantronics improved on the digipeater slightly and created KA-Nodes. As with digipeaters, KA-Nodes simply repeat AX.25 frames. However, a KA-Node acknowledges every transmission at each link (node) instead of over the entire route. Therefore, instead of an end-to-end acknowledgment, KA-Nodes allow for more reliable connections with fewer timeouts, because acknowledgments are only carried on one link.

"Netrom was one of the first networking schemes to try to address the problems with digipeaters. A user connects to a Netrom station as if connecting to any other packet station. From there, he can issue commands to instruct the station to connect to another user locally or connect to another Netrom station. This connect, then connect again, means that to a user's TNC, you are connected to a local station only and its transmissions do not have to be digipeated over the entire network and risk losing packets."

"NET/ROM is a commercial firmware (software put on a chip) program that is used as a replacement ROM in TAPR type TNCs. Other programs are available to emulate NET/ROM. Among them are TheNet, G8BPQ node switch, MSYS, and some versions of NET."

The key take-aways from the above is that using KA-Nodes and Netrom node results in:

- More reliable connections
- Fewer timeouts
- Less network traffic

However, with this new capability comes more complexity in configuring Outpost.

In general, to configure any node path, you need to perform the following steps:

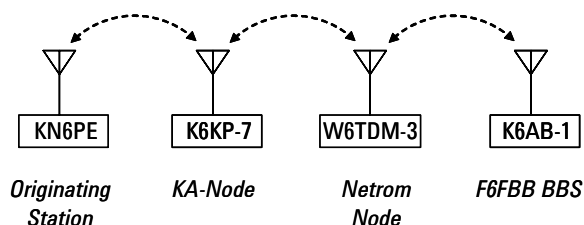
1. Collect information about the path to the BBS
2. Perform the connection manually before starting any Outpost setup
3. Record the relevant information for each node that you will need for the Outpost setup
4. Configure Outpost (and your BBS) for Node access
5. Testing the configuration

Example

The following example describes how to configure two types of nodes in Outpost.

Suppose there is an F6FBB BBS named K6AB-1 operating on 440.025 that I want to access, but cannot get to directly from 2 meters. However, I can reach it by going through 2 nodes that are between my station and the BBS. These stations are:

- K6KP: a dedicated Kantronics KPC-3 with its KA-Node feature turned on.
- W6TDM: a station running the BPQ Packet Switch software (a Netrom emulation)



Step 1. – Collect information about the path to the BBS

In our example, I know the following:

KN6PE	My home packet station that operates on 2 meters. I usually operate on 145.050.
K6KP-7	K6KP-7 is the connect name to get to this KA-Node. This node operates on 145.050 and is within range of my home station. Kantronics TNCs sets the default SSID for its KA-Nodes to “-7”, and appends it to the call sign entered with the Mycall command. The Node’s ID can be changed with the “ <u>mynode</u> <node_call_sign>” command.
W6TDM-3	W6TDM-3 is the connect name to get to the BPQ Packet Switch (Netrom) Node. The station owner configured this node with access name to have an “-3” SSID, and provides access to 3 ports that have radios on the following frequencies: <ul style="list-style-type: none"> ▪ Port 1 = 2 meters operating on 145.050 ▪ Port 2 = 440 band operating on 440.025 ▪ Port 3 = 220 band operating on 223.660
K6AB-1	K6AB-1 is the connect name to get to the BBS. This BBS operates on 440.025.

Step 2. – Manually connect to the BBS

This step confirms that you can really get to the BBS using the set of identified nodes.

I used Outpost’s serial terminal program (ipserial.exe) to execute these steps. However, other programs can be used, such as Hyperterm. Some additional lines are inserted into the listing below to improve readability.

Power up the TNC

This is the power-on message I get when turning on my TNC.

On power-on, the TNC’s startup message is displayed followed by the “cmd:” prompt.

Connect to the first node

Enter the command to connect to the first node in the path to the BBS.

If successful, there are 2 connect messages displayed.

- The first message, “*** CONNECTED ...”, is sent by my local TNC.
- The second message, “###CONNECTED...”, is sent from the K6KP KA-Node.

KA-Nodes are very friendly, and present a list of node commands that you could use.

Connect to the second node

While at the K6KP Node prompt, enter the command to connect to the second node in the path to the BBS.

Similar to the above, the “###LINK MADE” message is sent by the K6KP-7 KA-Node (1st node) after it connects to the 2nd Node.

```

KANTRONICS KPC3P VERSION 8.2
(C) COPYRIGHT 1997 BY KANTRONICS INC. ALL RIGHTS RESERVED.
DUPLICATION PROHIBITED WITHOUT PERMISSION OF KANTRONICS.
cmd:

cmd:c k6kp-7
cmd:*** CONNECTED to K6KP-7
###CONNECTED TO NODE K6KP-7(K6KP) CHANNEL A
ENTER COMMAND: B,C,J,N, or Help ?

c w6tdm-3
###LINK MADE
<cr>
?
CUPTNO:W6TDM-3} BBS CHAT CONNECT BYE INFO NODES PORTS
ROUTES USERS MHEARD
    
```

However, unlike the KA-Node, the BPQ node does not send back anything after you connect to it. Pressing “enter” only returns another blank line. Entering a “?” does produce the list of commands that the BPQ node does support. This is useful for manual interaction only.

Connect to the BBS

Finally, we enter the connect command to the BBS.

Because we started on 2 meters from my home, and the BBS is located on the 440 band, the Connect command is issued with an additional parameter to select port 2 (for the 440 frequency where the BBS operates). Once you connect to the BBS, it all looks familiar from here.

When done, enter a “b” for bye, and the disconnect process occurs all the way back to my local TNC command prompt.

```
c 2 k6ab-1
CUPTNO:W6TDM-3} Connected to K6AB-1
[FBB-7.04j-AB1FHMRX$]

Hello Jim,
Welcome to The San Jose EOC Mailbox

K6AB >
lm
*** : Nothing found
*** : TO Field Filter is set to:- [ * ]

K6AB >
b

73 Jim,
###DISCONNECTED BY W6TDM-3 AT NODE K6KP-7
*** DISCONNECTED
cmd:
```

Step 3. – Record the relevant information about the nodes

There are 4 things we need to know about every node that is in the path:

1. What is **connect** name for each node?

This is the station name of a node in the chain. It could be a call sign, call sign with SSID, or an alias. For example, Station Node Names can be:

K6FB-7: a KA-Node operating by the Las Cumbres Amateur Radio Club

W6TDM-3: a Netrom emulation running the G8BPQ Packet Switch software

DIA: the alias name of a station up on Mount Diablo in the San Francisco Bay Area

2. What message is displayed that indicates a **successful connection** was made to this node?

This is some or all of the message that the KA- or Netrom node will return when you successfully connect to this node. In the above example, the KPC-3 returned 2 lines of text indicating a successful connect. All we care about is the last line, and preferably the last couple of words on the line.

```
cmd:c k6kp-7
cmd:*** CONNECTED to K6KP-7
###CONNECTED TO NODE K6KP-7(K6KP) CHANNEL A
ENTER COMMAND: B,C,J,N, or Help ?
```

For the Netrom successful connect message, its fairly easy...

```
###LINK MADE
```

I've underlined the portions of the 2 strings above that Outpost can use. As with other aspects of Outpost, the program will be looking for these exact strings to know when it can proceed with sending another command.

3. What **port number** will be used to get to the next node or station in the path?

Some hardware and software packages that implement various packet network schemes have the ability to be configured with several radio ports (1, 2, 3, etc). In this case, The Port Number is the port that Outpost should use to get to the BBS or next node in the path. If you didn't enter a port number when you connected to the next station, the port number default should be "0".

4. What message is returned if an **unsuccessful connect** to the next node occurs?

Suppose we connected to the K6KP-7 KA-Node and are attempting to connect to the next node in the path. Assuming that the next node is not on line or you entered the node name incorrectly, we need to know the message that is returned from K6KP-7 indicating an unsuccessful connect. The best way to discover this is to force an error. For instance:

KA-Node

You connect to the first node as usual

```
cmd:c k6kp-7
cmd:*** CONNECTED to K6KP-7
###CONNECTED TO NODE K6KP-7(K6KP) CHANNEL A
ENTER COMMAND: B,C,J,N, or Help ?
```

You connect to the 2nd node. In this case, enter a station that does not exist.

```
c w6ztn-3
###RETRIED OUT AT NODE K6KP-7
ENTER COMMAND: B,C,J,N, or Help ?
```

The error message indicating a failed connect from the K6KP-7 node is the "###RETRIED" message.

Netrom Node

Connect to the 1st Node is successful

```
c k6kp-7
cmd:*** CONNECTED to K6KP-7
###CONNECTED TO NODE K6KP-7(K6KP) CHANNEL A
ENTER COMMAND: B,C,J,N, or Help ?
```

Connect to the 2nd node is successful

```
c w6tdm-3
###LINK MADE
```

Attempt to connect to the BBS, but with a station that does not exist.

```
c w6sjb-1
CUPTNO:W6TDM-3} Failure with W6SJB-1
```

In this case, an error message is returned. However, to keep the setup generic, I would only care about the phrase that excludes the specific station name that failed, in this case, W6SJB-1.

Once you have completed the above, I usually summarize the configuration so that I can easily recreate it later. For instance, to get from my home station (KN6PE) on 2 meters to the F6FBB BBS (K6AB-1) on 440, I recorded the following information that I will use to configure Outpost:

NODE #1

Node Name:	K6KP-7
Successful connect message to the node:	Help ?
Port number to get to the next node:	0
Unsuccessful connect message for next node:	###RETRIED

NODE #2

Node Name:	W6TDM-3
Successful connect message to the node:	###LINK MADE
Port number to get to the next node:	2
Unsuccessful connect message for next node:	Failure with

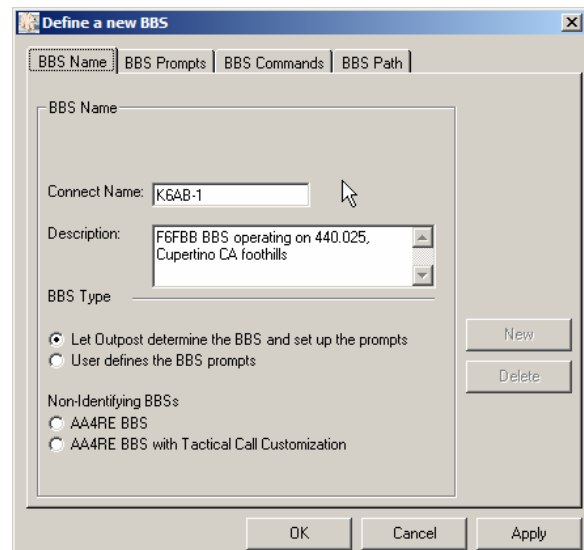
NOTES

1. For the K6KP-7 (KPC-3) Successful Connect Message, I'm only taking the last few characters since they are unique enough to indicate that we are connected.
2. For the W6TDM-3 (BPQ Packet Switch) Unsuccessful Connect Message, I'm taking a portion of the message that will be common enough so that this error message should work for other BBSs as well.
3. In all cases, be careful with spaces and capitalization... Outpost is looking for these literal strings.

Step 4. – Configure Outpost (and your BBS) for Node Access

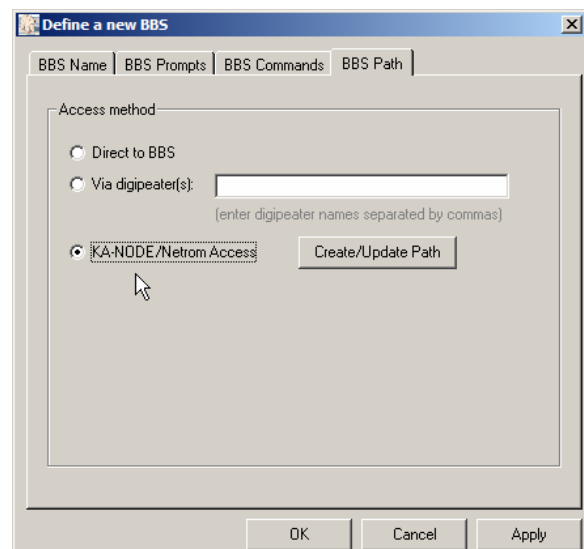
Finally, its time to set up the BBS in Outpost for node access.

1. From Outpost's main menu, select **Setup > BBS...**
On the 1st Tab, press **New**
Enter the BBS' connect name and any optional description.



2. On the 4th Tab, select the last option titled **KA-Node/Netrom Access**.

3. Press the **Create/Update Path** button to enter the information about the 2 nodes.



The Node Path Setup form allows you to add and delete nodes, then change the order by moving them up or down within the network path as necessary.

4. Enter the information for the 1st node... K6KP-7.
When done, press **Add**. The form is cleared and is ready for the next node.

5. Enter the information for the 2nd node... W6TDM-3.
When done, press **Next** or **Previous** to review the node entered.
NOTE: if the path became marginal and you needed to add another node somewhere in the path, you would add the node (it gets put at the end of the list) and then use the **Move Up** or **Move Down** buttons to position the new node in the right order.

When done, press **OK** to exit Node Path Setup, then press **OK** on the BBS setup form to save the changes.

Step 5. – Testing the Configuration

With all configurations entered, running Outpost will look exactly the same as if you entered all the commands manually. Here's the output of the run I did with the above configuration, with **Retrieve Private Messages**, **Retrieve New Bulletins**, and **Session Logging** turned on.

```
-----
15-Mar 10:47:34: Outpost v2.2.0 c194: TNC Session # 744
-----
```

```
cmd:b
BEACON EVERY 10 min
cmd:Echo on
ECHO was ON
cmd:my KN6PE
cmd:Mon off
MONITOR was OFF
cmd:daytime 061015123723
cmd:connect K6KP-7
cmd:*** CONNECTED to K6KP-7
###CONNECTED TO NODE K6KP-7(K6KP) CHANNEL A
ENTER COMMAND: B,C,J,N, or Help ?
connect W6TDM-3
###LINK MADE
connect 2 K6AB-1
CUPTNO:W6TDM-3} Connected to K6AB-1
[FBB-7.04j-AB1FHMRX$]

Hello Jim,
Welcome to the Cupertino Foothills Mailbox
```

```
K6AB >
LM
*** : Nothing found
*** : TO Field Filter is set to:- [ * ]

K6AB >
LB
*** : Nothing found
*** : TO Field Filter is set to:- [ * ]

K6AB >
B

73 Jim,
###DISCONNECTED BY W6TDM-3 AT NODE K6KP-7
*** DISCONNECTED
cmd:
```

Notes, Considerations, Limitations, etc.

The performance of the KA-Node/Netrom feature is dependent on a couple of things:

1. Quality of the configuration. Getting the path characteristics defined correctly will go a long way to ensure that Outpost will behave the way you expect. I encourage you to take the time to work through the configuration and confirm it all works for your environment before committing it within your operations.
2. Quality of the RF path. Users will always try to balance the number of nodes in a path to minimize excessive channel traffic with the need to have a guaranteed connection. Some of the best paths to BBSs that I experience during the day literally disappear during the night. Balancing RF path integrity with channel utilization is something you should consider when using KA-/Netrom Nodes (or even Digipeaters) in your operations.
3. Trading off robustness with complexity. I have had Outpost running KA-Nodes from home for weeks prior to releasing v2.1. By keeping the configuration relatively simple to implement, it does open the door for unexpected behavior and unhandled situations if a node along the path does not behave as anticipated. While I am hoping that the trade-offs between robustness and complexity were the right ones, I will defer to the user community to confirm that the mix is right.
4. Limitations. The only known limitation of the KA-Node implementation is with AGWPE connecting through a Netrom Node. For some unknown reason, the KA-Node I've used for test does not like the way AGWPE sends the subsequent connect request (works fine for serial access). This is an open issue and under investigation. If you are using AGWPE and this configuration is working for you, please let me know.