NAME

nghook - connect to a netgraph(4) node

SYNOPSIS

nghook [**-adlnSs**] [**-m** *msg*] *path* [*hookname*]
nghook -e [**-n**] [**-m** *msg*] *path hookname program* [*args* ...]

DESCRIPTION

The **nghook** utility creates a ng_socket(4) socket type node and connects it to hook *hookname* of the node found at *path*. If *hookname* is omitted, "debug" is assumed.

If the **-e** option is given, the third argument is interpreted as the path to a program, and this program is executed with the remaining arguments as its arguments. Before executing, the program Netgraph messages (specified by the **-m** option) are sent to the node. The program is executed with its standard input (unless closed by **-n**) and output connected to the hook.

If the **-e** option is not given, all data written to standard input is sent to the node, and all data received from the node is relayed to standard output. Messages specified with **-m** are sent to the node before the loop is entered. The **nghook** utility exits when EOF is detected on standard input in this case.

The options are as follows:

-a Output each packet read in human-readable decoded ASCII form instead of raw binary.

-d Increase the debugging verbosity level.

-e Execute the program specified by the third argument.

-l Loops all received data back to the hook in addition to writing it to standard output.

-m *msg*
Before executing the program (in **-e** mode) send the given ASCII control message to the node. This option may be given more than once.

-n Do not attempt to read any data from standard input. The **nghook** utility will continue reading from the node until stopped by a signal.

-S Use file descriptor 0 for output instead of the default 1.

-s Use file descriptor 1 for input instead of the default 0.

SEE ALSO

netgraph(3), netgraph(4), ngctl(8)

HISTORY

The **netgraph** system was designed and first implemented at Whistle Communications, Inc. in a version of FreeBSD 2.2 customized for the Whistle InterJet.

AUTHORS

Archie Cobbs <*archie@whistle.com*>

BUGS

Although all input is read in unbuffered mode, there is no way to control the packetization of the input.

If the node sends a response to a message (specified by **-m**), this response is lost.