NAME

pppoed - handle incoming PPP over Ethernet connections

SYNOPSIS

pppoed [**-Fd**] [**-P** *pidfile*] [**-a** *name*] [**-e** *exec* | **-l** *label*] [**-n** *ngdebug*] [**-p** *provider*] *interface*

DESCRIPTION

The **pppoed** utility listens to the given *interface* for PPP over Ethernet (PPPoE) service request packets, and actions them by negotiating a session then invoking a ppp(8) program. The negotiation is implemented by the "pppoe" netgraph node. See ng_pppoe(4) for details.

The **pppoed** utility will only offer services to clients requesting services from the given *provider*, which is taken as an empty name if not provided. If a provider name of "*" is given, any PPPoE requests will be offered service.

The supplied *name* will be given as the access concentrator name when establishing the connection. If no *name* is given, the current base hostname is used.

After receiving a request (PADI) from the PPPoE netgraph node, **pppoed** fork(2)s a child process and returns to service further requests. The child process offers service (using *name*) and waits for a SUCCESS indication from the PPPoE node. On receipt of the SUCCESS indication, **pppoed** will execute

exec */usr/sbin/ppp* **-direct** *label*

as a shell sub-process. If *label* has not been specified, it defaults to *provider*. It is possible to specify another command using the *exec* argument. This is mandatory if *provider* and *label* are not given. The child process will have standard input and standard output attached to the same netgraph(4) data socket (see ng_socket(4)) when started.

The environment variables HISMACADDR and ACNAME are made available to the child process and are set to the MAC address of the peer and the name of the AC respectively.

Upon invocation, **pppoed** will attach a "pppoe" netgraph node to the relevant "ether" node using "*interface*:" as the node name, and then connect that "pppoe" node to a local "socket" node. If the **-F** option has not been given, **pppoed** will then go into the background and disassociate itself from the controlling terminal. When the **-F** option is given, **pppoed** stays in the foreground.

If the **-d** option is given, additional diagnostics are provided (see the *DIAGNOSTICS* section below). If the **-n** option is given, **NgSetDebug**() is called with an argument of *ngdebug*.

If *pidfile* is given, **pppoed** will write its process ID to this file on startup.

DIAGNOSTICS

After creating the necessary netgraph(4) nodes as described above, **pppoed** uses syslogd(8) to report all incoming connections. If the **-d** option is given, **pppoed** will report on the child processes creation of a new netgraph socket, its service offer and the invocation of the ppp(8) program. If the **-n** option is given, netgraph diagnostic messages are also redirected to syslogd(8).

It is sometimes useful to add the following to */etc/syslog.conf*:

```
!pppoed
*.*                                     /var/log/pppoed.log
```

and the following to */etc/newsyslog.conf*:

```
/var/log/pppoed.log                     640  3     100    *     Z
```

SEE ALSO

NgSetDebug(3), netgraph(4), ng_ether(4), ng_pppoe(4), ng_socket(4), syslog.conf(5), ppp(8), syslogd(8)

HISTORY

The **pppoed** utility was written by Brian Somers <*brian@Awfulhak.org*> and first appeared in FreeBSD 3.4.

BUGS

If another netgraph node is using the given interface, **pppoed** will fail to start. This is because netgraph(4) does not currently allow node chaining. This may change in the future.