NAME

rfcomm_pppd - RFCOMM PPP daemon

SYNOPSIS

rfcomm_pppd **-c** [**-dh**] **-a** *address* **-C** *channel* **-l** *label* **-u** *N*
rfcomm_pppd **-s** [**-dDhS**] [**-a** *address*] **-C** *channel* **-l** *label*

DESCRIPTION

The **rfcomm_pppd** daemon is a simple wrapper daemon that allows the use of ppp(8) via an RFCOMM connection. It can operate in two modes: client and server.

In client mode, **rfcomm_pppd** opens an RFCOMM connection to the specified server’s *BD_ADRR* and *channel*. Once the RFCOMM connection is established, **rfcomm_pppd** executes ppp(8) in **-direct** mode with the specified *label*. Likewise, ppp(8) operates over the RFCOMM connection just like it would over a standard serial port, thus allowing a user to "dial out" and connect to the Internet.

In server mode, **rfcomm_pppd** opens an RFCOMM socket and listens for incoming connections from remote clients. Once the new incoming connection is accepted, **rfcomm_pppd** forks and executes ppp(8) in **-direct** mode with the specified *label*. Likewise, ppp(8) operates over the RFCOMM connection just like it would over a standard serial port, thus providing network connectivity to remote clients.

The options are as follows:

-a *address*

In client mode, this required option specifies the address of the remote RFCOMM server. In server mode, this option can be used to specify the local address to listen on. By default, in server mode, the daemon will listen on ANY address. The address can be specified as BD_ADDR or name. If a name was specified, the **rfcomm_pppd** utility will attempt to resolve the name via bt_gethostbyname(3).

-C *channel*

In both client and server mode, this required option specifies the RFCOMM channel to connect to or listen on. In server mode, the channel should be a number between 1 and 30. In client mode, the channel could either be a number between 1 and 30 or a service name. Supported service names are: **DUN** (Dial-Up Networking) and **LAN** (LAN Access Using PPP). If a service name is used instead of a numeric channel number, then **rfcomm_pppd** will try to obtain an RFCOMM channel number via SDP (Service Discovery Protocol).

-c Act as an RFCOMM client. This is the default mode.

-d Do not detach from the controlling terminal, i.e., run in foreground.

-D In server mode, register the **DUN** (Dial-Up Networking) service in addition to the **LAN** (LAN Access Using PPP) service. AT-command exchange can be faked with ppp(8) chat script.

-h Display usage message and exit.

-l *label*
In both client and server mode, this required option specifies which ppp(8) label will be used.

-S In server mode, register the **SP** (Serial Port) service in addition to the **LAN** (LAN Access Using PPP) service.

It appears that some cell phones are using the so-called "callback mechanism". In this scenario, the user is trying to connect his cell phone to the Internet, while the user’s host computer is acting as the gateway server. It seems that it is not possible to tell the phone to just connect and start using the **LAN** service. Instead, the user’s host computer must "jump start" the phone by connecting to the phone’s **SP** service. What happens next is the phone kills the existing connection and opens another connection back to the user’s host computer. The phone really wants to use the **LAN** service, but for whatever reason it looks for the **SP** service on the user’s host computer. This brain-damaged behavior was reported for the Nokia 6600 and the Sony/Ericsson P900.

-s Act as an RFCOMM server.

-u *N* This option maps directly to the **-unit** ppp(8) command-line option and tells **rfcomm_pppd** to instruct ppp(8) to only attempt to open */dev/tunN*. This option only works in client mode.

PPP CONFIGURATION

Important Notes on PPP Configuration

Special attention is required when adding new RFCOMM configurations to the existing PPP configuration. Please keep in mind that PPP will *always* execute commands in the "default" label of your */etc/ppp/ppp.conf* file. Please make sure that the "default" label *only* contains commands that apply to *every* other label. If you need to use PPP for both dialing out and accepting incoming RFCOMM connections, please make sure you have moved all commands related to dialing out from the "default" section into an appropriate outgoing label.

RFCOMM Server

One of the typical examples is the LAN access. In this example, an RFCOMM connection is used as a null-modem connection between a client and a server. Both client and server will start talking PPP right after the RFCOMM connection has been established.

```
rfcomm-server:
 set timeout 0
 set lqrperiod 10
 set ifaddr 10.0.0.1 10.0.0.2 255.255.255.0
 enable lqr
 accept lqr
 accept dns
 # Do not use PPP authentication. Assume that
 # Bluetooth connection was authenticated already
 disable pap
 deny pap
 disable chap
 deny chap
```

RFCOMM Client

The **rfcomm_pppd** utility supports both **LAN** (LAN Access Using PPP) and **DUN** (Dial-Up Networking) access. The client’s configuration for **LAN** access is very similar to the server’s and might look like this:

```
rfcomm-client:
 enable lqr
 accept lqr
 set dial
 set timeout 0
 disable iface-alias
 set ifaddr 10.0.0.1/0 10.0.0.2/0 255.255.255.0 0.0.0.0
 # Do not use PPP authentication. Assume that
 # Bluetooth connection was authenticated already
 deny pap
 disable pap
 deny chap
 disable chap
```

The client’s configuration for **DUN** access is different. In this scenario, the client gets connected to the virtual serial port on the server. To open a PPP session, the client must dial a number. Note that by default ppp(8) will not execute any configured chat scripts. The **force-scripts** option can be used to override this behavior. An example configuration is shown below:

```
rfcomm-dialup:
 # This is IMPORTANT option
 enable force-scripts

 # You might want to change these
 set authname
 set authkey
 set phone "*99***1#"

 # You might want to adjust dial string as well
 set dial "ABORT BUSY ABORT NO\\sCARRIER TIMEOUT 5 \
           \"\" AT OK-AT-OK ATE1Q0 OK \\dATDT\\T TIMEOUT 40 CONNECT"
 set login
 set timeout 30
 enable dns
 resolv rewrite

 set ifaddr 10.0.0.1/0 10.0.0.2/0 255.255.255.0 0.0.0.0
 add default HISADDR
```

Note that by adjusting the initialization string, one can make a CSD (Circuit Switched Data), HSCSD (High Speed Circuit Switched Data) or GPRS (General Packet Radio Service) connection. The availability of the particular connection type depends on the phone model and service plan activated on the phone.

EXIT STATUS

The **rfcomm_pppd** utility exits 0 on success, and >0 if an error occurs.

EXAMPLES

```
rfcomm_pppd -s -a 00:01:02:03:04:05 -C 1 -l rfcomm-server
```

This command will start **rfcomm_pppd** in the server mode. The RFCOMM server will listen on local address 00:01:02:03:04:05 and channel 1. Once the incoming connection has been accepted, **rfcomm_pppd** will execute ppp(8) in **-direct** mode with the "rfcomm-server" label.

```
rfcomm_pppd -c -a 00:01:02:03:04:05 -C 1 -l rfcomm-client
```

This command will start **rfcomm_pppd** in the client mode. **rfcomm_pppd** will try to connect to the RFCOMM server at 00:01:02:03:04:05 address and channel 1. Once connected, **rfcomm_pppd** will execute ppp(8) in **-direct** mode with the "rfcomm-client" label.

SEE ALSO

rfcomm_sppd(1), bluetooth(3), ng_btsocket(4), ppp(8), sdpcontrol(8), sdpd(8)

AUTHORS

Maksim Yevmenkin <*m_evmenkin@yahoo.com*>

CAVEATS

The **rfcomm_pppd** utility in server mode will try to register the Bluetooth LAN Access Over PPP service with the local SDP daemon. If the local SDP daemon is not running, **rfcomm_pppd** will exit with an error.