Huawei Module

Linux-based PPP Dial-up Connection Application Guide

Issue 01

Date 2016-04-26

Notice

Some features of the product and its accessories described herein rely on the software installed, capacities and settings of local network, and therefore may not be activated or may be limited by local network operators or network service providers.
Thus, the descriptions herein may not exactly match the product or its accessories which you purchase.
Huawei reserves the right to change or modify any information or specifications contained in this manual without prior notice and without any liability.

DISCLAIMER

ALL CONTENTS OF THIS MANUAL ARE PROVIDED “AS IS”. EXCEPT AS REQUIRED BY APPLICABLE LAWS, NO WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE MADE IN RELATION TO THE ACCURACY, RELIABILITY OR CONTENTS OF THIS MANUAL.
TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL HUAWEI BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES, OR LOSS OF PROFITS, BUSINESS, REVENUE, DATA, GOODWILL SAVINGS OR ANTICIPATED SAVINGS REGARDLESS OF WHETHER SUCH LOSSES ARE FORSEEABLE OR NOT.
THE MAXIMUM LIABILITY (THIS LIMITATION SHALL NOT APPLY TO LIABILITY FOR PERSONAL INJURY TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH A LIMITATION) OF HUAWEI ARISING FROM THE USE OF THE PRODUCT DESCRIBED IN THIS MANUAL SHALL BE LIMITED TO THE AMOUNT PAID BY CUSTOMERS FOR THE PURCHASE OF THIS PRODUCT.

Import and Export Regulations

Customers shall comply with all applicable export or import laws and regulations and be responsible to obtain all necessary governmental permits and licenses in order to export, re-export or import the product mentioned in this manual including the software and technical data therein.

Privacy Policy

To better understand how we protect your personal information, please see the privacy policy at http://consumer.huawei.com/privacy-policy.

About This Document

Revision History

Document Version	Date	Chapter	Description
01	2016-04-26		Initial release

Scope

MU509-b

MU509-c

MU509-65

MC509

MC323

MG323 series

MU609 series

MU709 series

Contents

1 Introduction

This document describes how Huawei modules use the pppd to initiate a Point-to-Point Protocol (PPP) dial-up connection on Linux operating systems (OSs, including the embedded Linux, computer-based Linux, and Android). It also provides guidance for developers who integrate PPP functions for Huawei modules to set PPP dial-up connection parameters.

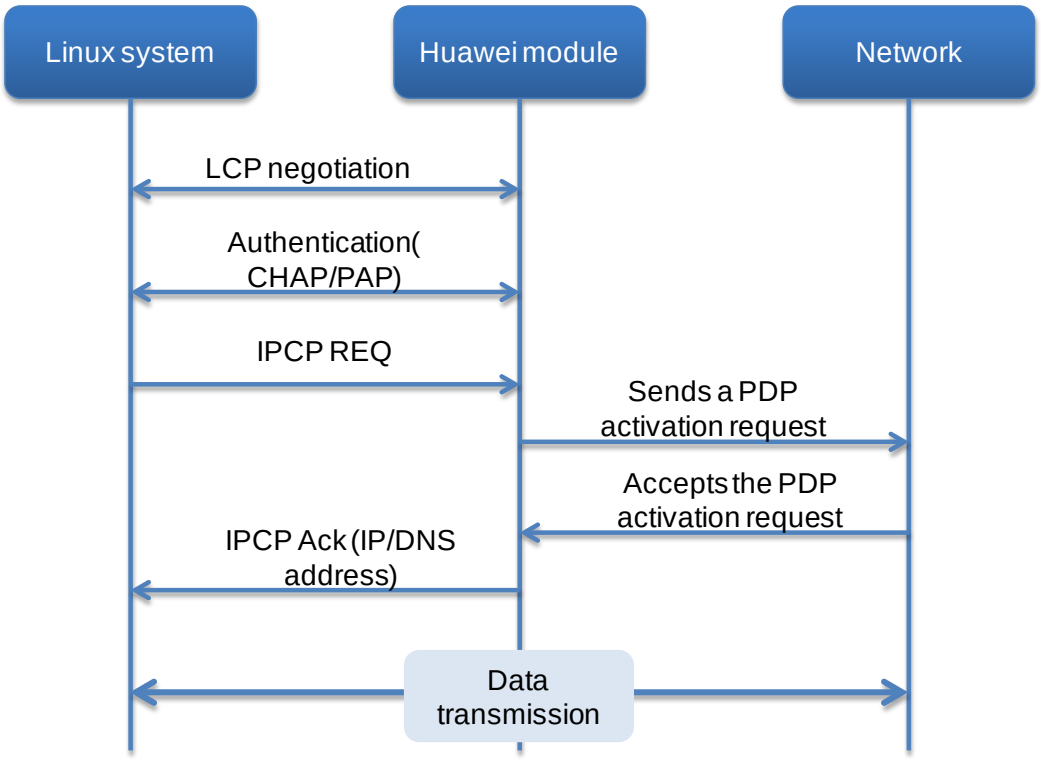
This document does not cover PPP dial-up connection configuration or setup on a graphical user interface (GUI).

2 PPP Dial-up Connection

On a Linux OS, setting up a PPP dial-up connection requires the pppd, where PPP dial-up parameters must be set in advance. The settings define how the pppd behaves and will affect PPP dial-up connections. Therefore, it is important to correctly configure these settings.

Figure 2-1 shows the PPP dial-up process.

Figure 2-1 PPP dial-up process



The following chapters describe the two methods to set the PPP dial-up parameters: using the command-line interface (CLI) to transmit parameters and using the script to set parameters. The former is recommended.

3 Use of the CLI for Parameter Transmission

3.1 AT Commands for Basic Configuration

3.1.1 3GPP-related Network Modes (GSM/WCDMA/TD-SCDMA)

1. On the AT command port (**/dev/ttyUSB2**), run the following command to set the APN:

   AT+CGDCONT=1,"IP","APN"

2. On the modem data port (**/dev/ttyUSB0**), run the following dial-up connection command:

   ATD*99#

   Or

   AT+CGACT=1,1

   AT+CGDATA=1

3.1.2 3GPP2-related Network Modes (CDMA 1X/EVDO)

1. (Optional) On the AT command port, run the following command to set the user name and password:

   AT^PPPCFG="username","password"

   To set the user name and password, you can also use the method described in section 3.2 "CLI Invoking and Parameter Setting".

2. (Required only when the network supports MIP) On the AT command port, run the following command to set the Session Initiation Protocol (SIP) or Mobile Internet Protocol (MIP) mode:

   AT^IPMODE=<mode>

   Currently, only the MC323 and MC509 Verizon editions support MIP.

   The values of <mode> are described as follows:

   - **0**: SIP only; the SIP dial-up process is similar to the PPP dial-up process on the WCDMA network.
   - **1**: MIP preferred; After a MIP dial-up connection fails to be set up, the module determines whether to switch to SIP based on the error code returned from the network.

- **2**: MIP only

3. On the modem data port (**/dev/ttyUSB0**), run the following dial-up connection command:

   ATD#777

NOTE

Before setting the **APN**, **username**, and **password** parameters in the preceding commands, consult with the local carrier, because a carrier's public network settings are different from its dedicated network settings.

3.2 CLI Invoking and Parameter Setting

In the CLI, execute the pppd and transmit the PPP dial-up parameters in command lines to the pppd. On some Linux systems, you may need to set the environment variables or add the absolute path of the pppd (**/system/bin/pppd** on the Android).

The commands provided in this section apply to common scenarios, which can be executed if no special requirements are imposed. Note that the settings of **authoption**, **USER**, and **PASSWD** must be consistent with those provided by the local carrier.

The following describes how to use the **authoption** parameter:

- **+PAP**/**+CHAP**/**+EAP**/**+** …: indicates that the Config-Request packet sent from the Linux system includes the PAP, CHAP, EAP, or other specified authentication option. In the pppd log shown in Figure 3-1 , the Config-Request packet sent from the Linux system includes the PAP option.

Figure 3-1 Function of the +PAP option

```
pppd options in effect:
debug    # (from command line)
nodetach    # (from command line)
dump    # (from command line)
+pap    # (from command line)
-chap    # (from command line)
user testqq    # (from command line)
password ??????    # (from command line)
/dev/ttyUSB244    # (from command line)
115200    # (from command line)
connect /system/bin/chat -v  -f /data/connect    # (from
disconnect    # (from command line)
nocrtscts    # (from command line)
novj    # (from command line)
noipdefault    # (from command line)
defaultroute    # (from command line)
usepeerdns    # (from command line)
Serial connection established.
using channel 45
Using interface ppp0
Connect: ppp0 <--> /dev/ttyUSB244
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <auth pap> <magi
debug:====sent  ff 03 c0 21 01 01 00 18 02 06 00 00 00 0
rcvd [LCP ConfReq id=0x60 <asyncmap 0x0> <auth chap MD5>
debug=======rcvd  ff 03 c0 21 01 60 00 19 02 06 00 00 00
```

- **-PAP/-CHAP/-EAP/**- ...: indicates that if the Config-Request packet sent from the module includes the PAP, CHAP, EAP, or other specified authentication option, the Linux system will respond with a Config-Nak packet to reject the specified authentication option. In the pppd log shown in Figure 3-2 , the Linux system does not process the Config-Request packet for the **-CHAP** option. However, the system includes the PAP option in the Config-Nak packet after detecting the CHAP option in the Config-Request packet from the module.

Figure 3-2 Function of the -CHAP option

```
pppd options in effect:
debug    # (from command line)
nodetach   # (from command line)
dump   # (from command line)
-chap    # (from command line)
-mschap    # (from command line)
-mschap-v2   # (from command line)
refuse-eap   # (from command line)
user wert3456$%567   # (from command line)
password ??????    # (from command line)
/dev/ttyUSB244   # (from command line)
115200   # (from command line)
connect /system/bin/chat -v  -f /data/connect    # (from command line)
disconnect     # (from command line)
nocrtscts    # (from command line)
novj   # (from command line)
noipdefault    # (from command line)
defaultroute   # (from command line)
usepeerdns   # (from command line)
Serial connection established.
using channel 128
Using interface ppp0
Connect: ppp0 <--> /dev/ttyUSB244
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0xa5d06caa> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0xa7 <asyncmap 0x0> <auth chap MD5> <magic 0x2d0c577> <pcc
debug=======rcvd  ff 03 c0 21 01 a7 00 19 02 06 00 00 00 00 03 05 c2 23 05 05 0
sent [LCP ConfNak id=0xa7 <auth pap>]
debug:====sent  ff 03 c0 21 03 a7 00 08 03 04 c0 23
rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0xa5d06caa> <pcomp> <accomp>]
debug=======rcvd  ff 03 c0 21 02 01 00 14 02 06 00 00 00 00 05 06 a5 d0 6c aa 0
rcvd [LCP ConfReq id=0xa8 <asyncmap 0x0> <auth pap> <magic 0x2d0c577> <pcomp> <
debug=======rcvd  ff 03 c0 21 01 a8 00 18 02 06 00 00 00 00 03 04 c0 23 05 06 0
sent [LCP ConfAck id=0xa8 <asyncmap 0x0> <auth pap> <magic 0x2d0c577> <pcomp> <
debug:====sent  ff 03 c0 21 02 a8 00 18 02 06 00 00 00 00 03 04 c0 23 05 06 02
```

Therefore, the meanings of **+PAP** and **-CHAP** are slightly different.

In general cases, you can set **authoption** according to the following table.

Expected Authentication Mode	Authoption Value
None	-PAP -CHAP
PAP	-CHAP
CHAP	-PAP
PAP&CHAP	N/A

For details about how to set other parameters, see section 4.2 "Options Script".

3.2.1 3GPP-related Network Modes (GSM/WCDMA/TD-SCDMA)

- If no user name and password are required, run the following in the shell CLI:

  pppd /dev/ttyUSB0 115200 mru 1280 nodetach debug dump defaultroute usepeerdns novj authoption novjccomp noipdefault ipcp-accept-local

ipcp-accept-remote connect-delay 5000 ipcp-max-failure 60 ipcp-max-configure 60 -am

- If the user name and password are required, run the following in the shell CLI:

  pppd /dev/ttyUSB0 115200 mru 1280 nodetach debug dump authoption defaultroute usepeerdns novj user *USER* password *PASSWD* novjccomp noipdefault ipcp-accept-local ipcp-accept-remote connect-delay 5000 ipcp-max-failure 60 ipcp-max-configure 60 -am

3.2.2 3GPP2-related Network Modes (CDMA 1X/EVDO)

- If no user name and password are required, run the following in the shell CLI:

  pppd /dev/ttyUSB0 115200 mru 1280 nodetach debug dump defaultroute usepeerdns novj authoption novjccomp noipdefault ipcp-accept-local ipcp-accept-remote connect-delay 5000 ipcp-max-failure 60 ipcp-max-configure 60

- If the user name and password are required, run the following in the shell CLI:

  pppd /dev/ttyUSB0 115200 crtscts debug dump authoption nodetach modem noipdefault defaultroute usepeerdns user *USER* password *PASSWD*

3.2.3 Example

Figure 3-3 Setting up a PPP dial-up connection using the CLI to transmit parameters

```
root@localhost:~# pppd /dev/ttyUSB0 115200 mru 1280 nodetach debug dump defaultroute usepeerdns
novj novjccomp noipdefault ipcp-accept-local ipcp-accept-remote connect-delay 5000
ipcp-max-failure 60 ipcp-max-configure 60
pppd options in effect:
debug   # (from command line)
nodetach    # (from command line)
connect-delay 5000    # (from command line)
dump    # (from command line)
/dev/ttyUSB0    # (from command line)
115200    # (from command line)
mru 1280    # (from command line)
novj    # (from command line)
novjccomp   # (from command line)
ipcp-accept-local   # (from command line)
ipcp-accept-remote    # (from command line)
noipdefault   # (from command line)
ipcp-max-configure 60   # (from command line)
ipcp-max-failure 60   # (from command line)
defaultroute    # (from command line)
usepeerdns    # (from command line)
using channel 15
Using interface ppp0
Connect: ppp0 <--> /dev/ttyUSB0
sent [LCP ConfReq id=0x1 <mru 1280> <asyncmap 0x0> <magic 0xb937e701> <pcomp> <accomp>]
rcvd [LCP ConfReq id=0x3 <asyncmap 0x0> <auth chap MD5> <magic 0x6803e68f> <pcomp> <accomp>]
sent [LCP ConfAck id=0x3 <asyncmap 0x0> <auth chap MD5> <magic 0x6803e68f> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x1 <mru 1280> <asyncmap 0x0> <magic 0xb937e701> <pcomp> <accomp>]
rcvd [LCP DiscReq id=0x4 magic=0x6803e68f]
rcvd [CHAP Challenge id=0x1 <2c646d5176aecbe4523a2bf8c65c02de>, name = "UMTS_CHAP_SRVR"]
sent [CHAP Response id=0x1 <3c2d6b609e611d1b324e55648cacca9f>, name = "localhost"]
rcvd [CHAP Success id=0x1 ""]
CHAP authentication succeeded
CHAP authentication succeeded
sent [CCP ConfReq id=0x1 <deflate 15> <deflate(old#) 15> <bsd v1 15>]
sent [IPCP ConfReq id=0x1 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
rcvd [LCP ProtRej id=0x5 80 fd 01 01 00 0f 1a 04 78 00 18 04 78 00 15 03 2f]
Protocol-Reject for 'Compression Control Protocol' (0x80fd) received
rcvd [IPCP ConfReq id=0x2]
sent [IPCP ConfNak id=0x2 <addr 0.0.0.0>]
rcvd [IPCP ConfNak id=0x1 <addr 192.168.70.41> <ms-dns1 172.22.44.200> <ms-dns2 172.22.45.201>]
sent [IPCP ConfReq id=0x2 <addr 192.168.70.41> <ms-dns1 172.22.44.200> <ms-dns2 172.22.45.201>]
rcvd [IPCP ConfReq id=0x3]
sent [IPCP ConfAck id=0x3]
rcvd [IPCP ConfAck id=0x2 <addr 192.168.70.41> <ms-dns1 172.22.44.200> <ms-dns2 172.22.45.201>]
Could not determine remote IP address: defaulting to 10.64.64.64
not replacing existing default route via 10.11.38.1
local  IP address 192.168.70.41
remote IP address 10.64.64.64
primary   DNS address 172.22.44.200
```

In Figure 3-3 :

- The log information enclosed in a red frame indicates the commands executed and parameters transmitted.
- The log information enclosed in a green frame indicates the pppd's parsing of the parameters.
- The log information enclosed in a blue frame indicates the process from the negotiation between the pppd and module until the IP address is obtained.

4 PPP Dial-up Script Configuration

This chapter provides simple instances to describe how to use scripts to set PPP dial-up parameters and set up a PPP dial-up connection. For details about the parameter setting, obtain the Huawei PPP dial-up script from Huawei's module technical personnel.

The two scripts related to PPP dial-up are chat and options. The chat script is used to set PPP dial-up related AT commands, while the options script is used to set PPP dial-up related parameters.

4.1 Chat Script

4.1.1 Simple Chat Script

The structure of a simple chat script is as follows:

```
""          AT
OK          ATDT    dialnumber
CONNECT     ""
```

The chat script consists of string pairs. The string on the left is the expected string, while the string on the right is the command to send. If the expected string is not received, the string on the right will not be sent.

The following describes the strings in the script structure:

- The first line indicates that the module will send the string "AT" regardless of the string it receives.
- The second line indicates that the module will send the string **"ATDT dialnumber"** only when it receives the string **"OK"**. The value of **dialnumber** varies by network mode. For example, the value is *99# for WCDMA and #777 for CDMA.
- The third line indicates that the module will not send any more strings when it receives the string **"CONNECT"** because the module considers the data link as having been set up.

4.1.2 Typical Chat Script

- If timeout control is required, you can add **TIMEOUT 10**, whose value unit is second.
- If handling of special cases is required, add the following fields:

  ABORT BUSY
  ABORT NO ANSWER
  ABORT RINGING

  The preceding statements indicate that the module will exit execution if it receives the string **"BUSY"**, **"NO ANSWER"**, or **"RINGING"**.

Considering all the special cases, a chat script can be configured as follows:

TIMEOUT 30
ABORT BUSY
ABORT NO ANSWER
ABORT RINGING
"" AT
OK ATDT dialnumber
CONNECT ""

4.2 Options Script

The options script can specify the device used for the PPP dial-up connection, string transmission speed, hardware acceleration, overflow, and more.

The followings are some parameters in the options script:

- **ttyS0**: specifies the modem port number used for the PPP dial-up connection. For Huawei modules, the modem port is **ttyUSB0** in most cases.
- **57600**: specifies the baud rate (bit/s) used in the PPP dial-up connection. For Huawei modules, it is recommended that you set this parameter to **115200**.
- **debug**: indicates that debugging is required.
- **logfile /var/ ppplog**: specifies the file to which the information generated during connection setup will be exported.
- **mtu 1500**
- **-detach**
- **noipdefault**: specifies whether a parameter is not used by default.
- **defaultroute**
- **usepeerdns**: specifies the DNS server negotiated by the server.
- **lcp-echo-failure 4**: indicates that the module considers the server as no responding if it does not receive any LCP request responses for four consecutive times. The number of failures is configurable.
- **-ccp**: indicates that the Compression Control Protocol is not used.
- **-vj**: disables IP Header Compression.
- **-chap**: disables CHAP.

- **-mschap-v2**: disables MSCHAP.
- **user**
- **hide-password**
- **connect "/usr/bin/chat -v -t6 -f /var/ chat"**: specifies the chat script location. **-v** indicates that all chat command input and output will be copied to the system records (the save path is **/var/log/messages** in most cases). **-t6** indicates that the chat command execution time is 6s. The chat script can be saved to **/etc/** or **/var**.
- **persist**: indicates that the module will automatically redial in the case of dial-up failures.
- **crtscts**: asks the pppd to use the modem's hardware-based flow control.
- **modem**: asks the pppd to use DCD signals to determine whether a connection is working properly.
- **deflate**: asks the pppd to use the defalte compression mode.
- **idle**: specifies a time limit. If no data is transmitted before this limit is exceeded, the connection will be terminated.
- **lock**: creates a lock file to announce to other applications that the modem port specified by **ttyS0** is already in use.
- **demond**: indicates that the pppd will always run in the background and monitor the network data. The pppd will set up a dial-up connection when required and terminate the connection when the time limit is exceeded.

For details about the other parameters, see the following, which are the options.c notes in the pppd 2.4.4 source code document:

```
/*
 * Option variables and default values.
 */
int debug = 0;    /* Debug flag */
int kdebugflag = 0;    /* Tell kernel to print debug messages */
int default_device = 1; /* Using /dev/tty or equivalent */
char devnam[MAXPATHLEN]; /* Device name */
bool nodetach = 0;    /* Don't detach from controlling tty */
bool updetach = 0;    /* Detach once link is up */
int maxconnect = 0;    /* Maximum connect time */
char user[MAXNAMELEN]; /* Username for PAP */
char passwd[MAXSECRETLEN]; /* Password for PAP */
bool persist = 0;    /* Reopen link after it goes down */
char our_name[MAXNAMELEN]; /* Our name for authentication purposes */
bool demand = 0;    /* do dial-on-demand */
char *ipparam = NULL; /* Extra parameter for ip up/down scripts */
int idle_time_limit = 0; /* Disconnect if idle for this many seconds */
```

```
int holdoff = 30;    /* # seconds to pause before reconnecting */
bool holdoff_specified; /* true if a holdoff value has been given */
int log_to_fd = 1;    /* send log messages to this fd too */
bool log_default = 1; /* log_to_fd is default (stdout) */
int maxfail = 10;    /* max # of unsuccessful connection attempts */
char linkname[MAXPATHLEN]; /* logical name for link */
bool tune_kernel;    /* may alter kernel settings */
int connect_delay = 1000; /* wait this many ms after connect script */
int req_unit = -1;    /* requested interface unit */
bool multilink = 0;    /* Enable multilink operation */
char *bundle_name = NULL; /* bundle name for multilink */
bool dump_options;    /* print out option values */
bool dryrun;      /* print out option values and exit */
char *domain;    /* domain name set by domain option */
int child_wait = 5;    /* # seconds to wait for children at exit */
```

4.3 Authentication Script

In general cases, a PPP dial-up connection requires identity authentication. Two authentication types are mainly used: PAP and CHAP. The user name and password used for authentication are saved in the **PAP-secrets** or **chap-secrets** script, depending on the authentication type.

```
username * password
```

These scripts are generally placed under **/etc/ppp/** on the Linux system. If they are unavailable, manually create one.

If authentication is required, specify the authentication type (PAP or CHAP) in the options script. The PPP module will then read the user name and password from the **PAP-secrets** or **chap-secrets** script and add the user name and password to the PPP authentication package. Then, the module sends the package to the server for identity authentication.

4.4 Relationship Between the PPP Dial-up Process and Scripts

After the script is successfully configured, it can work with the pppd as follows:

1. The pppd locates the chat script based on the options script, and execute AT commands based on the chat script.

2. The pppd sets the other parameters based on the options script to set up a PPP dial-up connection.

4.5 Example

The example provided in this section gives the options, chat, and PAP-secrets scripts used to set up a PPP dial-up connection on the MU509-b module.

Using these scripts, a PPP dial-up connection can be successfully set up.

4.5.1 Options Script

connect "/usr/bin/chat -v -t6 -f /etc/ppp/my_chat"

ttyUSB0

115200

debug

logfile /var/log/ppplog

mtu 1500

-detach

noauth

noipdefault

defaultroute

usepeerdns

crtscts

lock

lcp-echo-failure 4

-ccp

-vj

-chap

-mschap-v2

user

hide-password

4.5.2 Chat Script

"" AT

OK AT+CGDCONT=1,"IP","3gnet"

OK ATD*99#

CONNECT ""

4.5.3 PPAP-secrets Script

wap *wap

4.5.4 PPP Dial-up Operation

In the CLI, invoke the pppd, and transmit the options script location and file name to the pppd as CLI parameters.

 # pppd call ./mydialerup

5 Acronyms and Abbreviations

Acronym or Abbreviation	Expansion
APN	Access Point Name
CHAP	Challenge Handshake Authentication Protocol
PAP	Password Authentication Protocol
PPP	Point-to-Point Protocol
USB	Universal Serial Bus