

Skywire® LTE NL-SW-LTE-SRC7611 Modem Family

Linux Networking Guide

NimbeLink Corp.

Updated: March 2024



© NimbeLink Corp. 2024. All rights reserved.

NimbeLink Corp. provides this documentation in support of its products for the internal use of its current and prospective customers. The publication of this document does not create any other right or license in any party to use any content contained in or referred to in this document and any modification or redistribution of this document is not permitted.

While efforts are made to ensure accuracy, typographical and other errors may exist in this document. NimbeLink reserves the right to modify or discontinue its products and to modify this and any other product documentation at any time.

All NimbeLink products are sold subject to its published Terms and Conditions, subject to any separate terms agreed with its customers. No warranty of any type is extended by publication of this documentation, including, but not limited to, implied warranties of merchantability, fitness for a particular purpose and non-infringement.

NimbeLink and Skywire are registered trademarks of NimbeLink Corp. All trademarks, service marks and similar designations referenced in this document are the property of their respective owners.

Table of Contents

Table of Contents	2
1. Introduction	4
1.1 Scope	4
1.2 Overview	4
1.3 Orderable Part Numbers	5
2. Preliminary Setup Procedure	5
2.1 Linux PC Setup	5
2.1.1 Notes	5
2.1.2 Elevate to Root	5
2.1.3 Check for Package Updates	5
2.1.4 Install Required Packages	5
2.1.5 Install Optional Packages	6
2.1.6 Linux Namespaces	6
2.2 Skywire Modem Setup	8
2.2.1 Prepare the Hardware	8
2.2.2 Verify Modem Enumeration	8
2.2.3 Load the "option" Driver (If Needed)	9
3. Point-to-Point Protocol (PPP)	10
3.1 Overview	10
3.2 PPP Scripts	10
3.2.1 NimbeLink PPP Script GitHub Repository	10
3.2.2 Create PPP Scripts	10
3.2.3 Edit the "x-SRC7611-chat" Script	11
3.2.4 Edit the "x-SRC7611" Script	12
3.3 Choose a PPP Implementation	12
3.4 Standard PPP Procedure	13
3.4.1 Take down Network Interfaces	13
3.4.2 Start the PPP Session	13
3.4.3 Test the PPP Session	14
3.4.4 Close the PPP Connection	14
3.5 Alternate Procedure for Testing PPP	14

3.5.1	Create a Network Namespace	15
3.5.2	Start the PPP Session	15
3.5.3	Test the PPP Connection	15
3.5.4	Close the PPP Connection	17
3.5.5	Delete the Linux Namespace	17
3.6	PPP Troubleshooting	18
4.	QMI	19
4.1	System Set Up	19
4.2	Using QMI	19
4.2.1	Disconnect Other Interfaces	19
4.2.2	Gather Information	19
4.2.3	Create Configuration File	20
4.2.4	Start QMI	20
4.2.5	Configure Networking	21
4.2.6	Using DNS	23
4.3	Troubleshooting	24
5.	Document Version Information	25

1. Introduction

1.1 Scope

This document serves as a networking guide for NL-SW-LTE-SRC7611 Skywire modems and Linux host systems running Debian-based operating systems.

1.2 Overview

This application note covers two networking protocols for Linux environments: PPP and QMI.

All protocols allow Skywire modems to provide an Internet connection when connected over USB. The Internet connection provided by the modem behaves similarly to an Ethernet connection, so this networking technique is often called "Ethernet over USB".

For connections over UART, PPP can be used.

[Section 2](#) covers preliminary steps that should be performed, including: checking for package updates, downloading new packages, powering on the modem, and verifying modem enumeration on the Linux PC. Follow the instructions in this section before trying the following.

Linux users can choose from PPP and QMI protocols when establishing an Ethernet over USB connection:

- [Section 3](#): Point-to-Point Protocol (PPP)
- [Section 4](#): Qualcomm Mobile Station Modems (MSM) Interface (QMI)

Note: *QMI generally has higher download and upload speeds than PPP.*

The networking methods described in this guide have been tested on the following Linux distributions:

- Ubuntu 18.04 LTS
- Ubuntu 20.04 LTS
- Debian 9.5 2018-10-07

1.3 Orderable Part Numbers

Orderable Device	Cellular Carrier	Network Type
NL-SW-LTE-SRC7611-4	AT&T, Verizon	LTE
NL-SW-LTE-SRC7611-4NG-B	AT&T, Verizon	LTE
NL-SWDK	Any	Any

2. Preliminary Setup Procedure

2.1 Linux PC Setup

2.1.1 Notes

The commands in this guide are for the Debian and Ubuntu listed in Section 1.2. If you are using a different Linux distribution, please modify the commands as needed.

2.1.2 Elevate to Root

To make the process easier, it is best to elevate to root before continuing with this guide. To do so, open a Linux terminal, and issue the following command:

```
sudo -i
```

2.1.3 Check for Package Updates

Ensure that host system has the most recent versions of installed packages by issuing the following command:

```
apt update && apt upgrade
```

This command will check for package updates, and then perform any updates it finds.

2.1.4 Install Required Packages

Additional packages may need to be installed on the Linux PC, depending on the networking protocol(s) that will be tested:

- **PPP**
 - pppd - Point-to-Point Protocol Daemon.
 - Install with: `apt install ppp`

- The "ppp" package adds a background process that handles PPP traffic alongside the kernel's PPP driver. This is known as a PPP daemon (PPPd).
- For more information, see the following man page:
<https://manpages.debian.org/testing/ppp/pon.1.en.html>

- **QMI**

- Install with: **sudo apt install libqmi-utils**
- For more information, see the following man page:
<https://manpages.debian.org/unstable/libqmi-utils/qmicli.1.en.html>

2.1.5 Install Optional Packages

Depending on the chosen method of communication between the Linux PC and the Skywire, additional package installations may be required:

- **picocom** – minimal Dumb-Terminal Emulation Program
- Install with: **apt install picocom**
- NimbeLink recommends using picocom to handle serial and USB communication between the modem and the Linux host PC.
- For more information, see the following man page:
<https://linux.die.net/man/8/picocom>
- Any similar package, like minicom, can be used instead of picocom, if desired.

2.1.6 Linux Namespaces

In addition to the standard procedures for PPP, this application note has alternate instructions that make use of Linux Namespaces.

A namespace is a container within the operating system that allows for certain processes or resources to be isolated from the rest of the system. Specifically, a Network namespace can be used to isolate the Ethernet-over-USB connection from any other network interfaces that may be present on the host PC.

This approach is helpful while testing an implementation of the Skywire, and can be advantageous for designers that use SSH or Telnet to communicate with the Linux PC. Since the standard PPP procedures involve disabling network interfaces, the Linux PC loses its ability to communicate over Ethernet. By isolating the Ethernet over USB connection using a Linux Namespace, the problem of having to take down the Ethernet interface on the Linux PC is avoided during testing.

Note: *It is possible to leave network interfaces enabled on the Linux PC, while ensuring that the cellular data connection is the primary source of Internet connectivity. This involves replacing the default route in the kernel's IP routing table with the PPP or QMI connection. However, these steps are not covered by this application note.*



The alternate methods involving Linux Namespaces are meant for testing and prototyping use only. It is not recommended to utilize Linux Namespaces in the production implementation of an Ethernet over USB connection. Instead, follow the standard procedure(s) when designing an implementation for production devices.

2.2 Skywire Modem Setup

2.2.1 Prepare the Hardware

Ensure that the modem has an activated SIM card inserted into the SIM slot, or that the modem's soldered-down SIM is active.

Power on the modem and allow it to boot up. Using a Linux terminal or another preferred method, open a serial or USB connection to the Skywire.

- **SWDK Users:** Refer to the SWDK user manual for detailed setup instructions:

https://nimbelink.com/Documentation/Development_Kits/NL-SWDK/30005_NL-SWDK_UserManual.pdf

- Ensure that the active firmware image corresponds to the SIM card in use. Section 3.9 of the NL-SW-LTE-SRC7611-4 datasheet contains information regarding the dual image firmware:

https://nimbelink.com/Documentation/Skywire/4G_LTE_Cat_4_DualCarrier/1002884_NL-SW-LTE-SRC7611_Datasheet.pdf

2.2.2 Verify Modem Enumeration

After booting the modem, verify that the modem has properly enumerated on the Linux PC. Type the following command into the Linux terminal:

```
lsusb
```

If the Skywire's USB interface has properly enumerated on the Linux PC, there should be a device listed that looks something like:

```
Bus 001 Device 02: ID 1199:[PID] Sierra Wireless, Inc.
```

Where "[PID]" is replaced with the PID that corresponds to the active USB interface composition.

If a line similar to the one above appears, then the modem has properly enumerated on the Linux PC.

If a USB device does not appear, verify that the modem is powered on and that the USB interface is connected to the Linux PC. Then, retry the "**lsusb**" command.

If the modem still has not enumerated as expected on the Linux PC, try the troubleshooting step in the next section.

2.2.3 Load the “option” Driver (If Needed)

On older versions of Ubuntu and other Linux distributions, newer Skywire modems will sometimes fail to enumerate fully, if at all.

If the modem does not enumerate on the host PC, or if some of the modem's USB ports are missing from the Linux PC, the "option" driver must be loaded for the modem. Issue the following commands:

```
modprobe option
```

```
echo 1199 [PID] > /sys/bus/usb-serial/drivers/option1/new_id
```

Be sure to replace "[PID]" with the PID that corresponds to the active USB composition for the Skywire.

In the case of an NL-SW-LTE-SRC7611-4 Skywire with the default USB composition, the above command would be:

```
echo 1199 68c0 > /sys/bus/usb-serial/drivers/option1/new_id
```

After issuing the above command, the Skywire's USB interface should now fully enumerate on the Linux PC.

At this point, the Linux PC and the Skywire are ready for the Ethernet over USB connection. [Section 3](#) contains instructions for PPP and [Section 4](#) contains instructions for QMI.

Note: *If the modem still does not enumerate, then it is possible that the USB interface is damaged on the modem. For further assistance, please contact the following email:*

product.support@nimbelink.com

3. Point-to-Point Protocol (PPP)

3.1 Overview

Point-to-Point Protocol (PPP) is a communications protocol used to directly connect two devices together without any other networking devices placed in between. PPP is a data link layer protocol, and is designed to work with multiple network layer protocols, such as Internet Protocol (IP).

Section 3 describes how to set up PPP on an NL-SW-LTE-SRC7611-4 Skywire modem and a Linux host system like a PC, or a single-board computer (BeagleBone, Raspberry Pi, etc.).

Note: These instructions assume that the Linux host PC and the Skywire have both been set up according to [Section 2](#), and that the "pppd" package has been installed on the PC.

3.2 PPP Scripts

3.2.1 NimbeLink PPP Script GitHub Repository

NimbeLink has a library of PPP scripts available for customer use at the following link:

<https://github.com/NimbeLink/skywire-ppp-scripts>

NimbeLink recommends downloading this repository, and following the instructions on the GitHub page to set up the scripts.

To clone the repository, navigate to `"/etc/ppp/peers"`, and enter the following command into the Linux terminal:

```
git clone https://github.com/NimbeLink/skywire-ppp-scripts.git
```

Alternatively, the contents of the PPP scripts can be copied directly from an Internet browser into the proper location on the Linux PC's filesystem. However, it is recommended to simply clone the repository.

3.2.2 Create PPP Scripts

As root user, navigate to the `"/etc/ppp/peers"` directory on the Linux host PC. This directory is where all of the PPP scripts for Skywire modems should be placed.

Next, identify the pair of PPP scripts that correspond to the carrier of the SIM card in use, and then copy those scripts to the `"/etc/ppp/peers"` directory.

The commands below demonstrate this operation, assuming that the PPP script repository was cloned into the home directory of the root user. Change the path to the `"skywire-ppp-scripts"` repository in the command below, if applicable:

Verizon Users:

```
cp ~/skywire-ppp-scripts/vzw-SRC7611* /etc/ppp/peers
```

AT&T Users:

```
cp ~/skywire-ppp-scripts/att-SRC7611* /etc/ppp/peers
```

To Copy Both AT&T and Verizon Scripts:

```
cp ~/skywire-ppp-scripts/*SRC7611* /etc/ppp/peers
```

If the copying operation was done correctly, the NL-SW-LTE-SRC7611-4 PPP scripts should now be in the `"/etc/ppp/peers"` directory.

Verify this by comparing the contents of the `"/etc/ppp/peers"` with the list of required PPP scripts below:

Required Verizon PPP Scripts:

- `"vzw-SRC7611-chat"`
- `"vzw-SRC7611"`

Required AT&T PPP Scripts:

- `"att-SRC7611-chat"`
- `"att-SRC7611"`

For example, the image below depicts the contents of the `"/etc/ppp/peers"` directory with the two required AT&T scripts:

```
root@blackberrypi:/etc/ppp/peers# ls -l
total 44
-rw-r--r-- 1 root dip 174 Aug 31 16:17 att-bg96
-rw-r--r-- 1 root dip 149 Sep 1 09:09 att-bg96-chat
-rw-r--r-- 1 root dip 135 Sep 8 12:19 att-SRC7611
-rw-r--r-- 1 root dip 137 Sep 8 11:01 att-SRC7611-chat
```

3.2.3 Edit the `"x-SRC7611-chat"` Script

Before using the scripts, the `"x-SRC7611-chat"` file must be edited to add the proper APN for the cellular connection. Replace `"x"` in the filename with either `"att"` or `"vzw"`, depending on the SIM card being used.

Open the `"x-SRC7611-chat"` file with a preferred text editor, and find the line that says:

AT&T Users:

- `OK AT+CGDCONT=1,"IP", "[apn]"`

Verizon Users:

- OK AT+CGDCONT=3,"IP","[apn]"

Replace "[apn]" with the proper APN for the chosen SIM card and cellular carrier.

For example, the image below depicts an edited "att-SRC7611-chat" file with the proper APN set for an AT&T SIM card purchased through NimbeLink:

```
TIMEOUT 35
ECHO ON
' ' \rATZ
OK 'ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0'
OK AT+CGDCONT=1,"IP","iot0718.com.attz"
OK AT+CEREG?;+CESQ;+CGDCONT?;+CFUN?
OK ATD*99***1#
CONNECT ' '
```

3.2.4 Edit the "x-SRC7611" Script

Next, the "x-SRC7611" file may need to be updated to reflect the proper port for the PPP session. Replace "x" in the filename with either "att" or "vzw", depending on the SIM card being used.

Open this file with a preferred text editor, and change the line that says "/dev/ttyUSB3" to reflect the port to which the modem enumerated (if necessary).

For instance, if the modem enumerated to the "/dev/ttyACM0" port, this port should be specified in the "x-SRC7611" file.

3.3 Choose a PPP Implementation

After the PPP scripts have been copied and edited as described above, they are ready to be used for the PPP session. Choose one of the following sections to continue:

[Section 3.4](#): Standard PPP Procedure

- **This is the standard PPP setup procedure that should be used for production implementations of PPP.**
- This method also works well for testing, provided that the user is not using SSH or Telnet to communicate with the Linux PC over Ethernet.

[Section 3.5](#): Alternate PPP Procedure for Testing Purposes

- **This is an alternate method for setting up PPP meant for testing only. Do not use this method for production implementations of PPP.**

- This method is useful for readers who use SSH or Telnet to communicate with the Linux PC over Ethernet, because it won't sever that communication path like the standard PPP procedure does.

Additionally, [Section 3.6](#) contains troubleshooting tips that may be useful when attempting to start the PPP session.

3.4 Standard PPP Procedure

3.4.1 Take down Network Interfaces

Note: *This step will sever the communication path over Ethernet with the Linux PC.*

Issue the following command in the Linux terminal:

```
ifconfig
```

If needed, install with **sudo apt install net-tools**.

This command lists all currently active network interfaces. Take note of each entry that is populated in the response to the above command.

Disable each interface that may provide an Internet connection to the Linux PC. Issue the following command to take each interface down:

```
ifconfig [interface] down
```

Replace "[interface]" with the name of the interface that is to be disabled. Common network interfaces are: **eth#**, **enp##**, **wlan#**, **wwan#**, etc.

Once the network interfaces have been disabled, check the PC's Internet connectivity using the Linux terminal:

```
ping 8.8.8.8 -c 5
```

This will cause the Linux PC to attempt to ping Google's public DNS server five times. If all of the relevant network interfaces have been disabled, the ping attempt will fail.

If the ping attempt succeeds, then verify that all relevant network interfaces have been disabled with the "**ifconfig [interface] down**" command.

Once all sources of Internet connectivity have been disabled, the PPP session can be tested properly.

3.4.2 Start the PPP Session

Start the PPP session with the following command:

```
pon [ppp script]
```

Replace "[ppp script]" with one of the following filenames, depending on the carrier of the chosen SIM card:

Verizon Users: "vzw-SRC7611"

AT&T Users: "att-SRC7611"

This command will start the pppd daemon, which will configure the PPP connection, and set it as the default network connection.

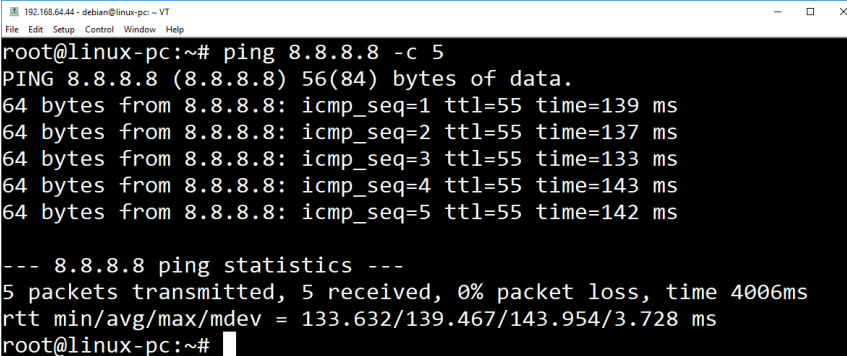
Note: Ensure that the modem does not have an active PDP context when starting the PPP script. This will cause the PPP script to encounter an error and freeze up.

3.4.3 Test the PPP Session

To test that the PPP connection is working properly, issue the ping command once more:

```
ping 8.8.8.8 -c 5
```

If the PPP session was successful, the response should look something like:



```
192.168.64.44 - debian@linux-pc: ~ - VT
File Edit Setup Control Window Help
root@linux-pc:~# ping 8.8.8.8 -c 5
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=55 time=139 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=55 time=137 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=55 time=133 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=55 time=143 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=55 time=142 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 133.632/139.467/143.954/3.728 ms
root@linux-pc:~#
```

3.4.4 Close the PPP Connection

To close the PPP connection, simply issue this command:

```
poff
```

It is a good idea to close a PPP session whenever it is no longer needed. This will prevent accidental usage of cellular data.

Note: This concludes the standard implementation of PPP for NL-SW-LTE-SRC7611 Skywire modems.

3.5 Alternate Procedure for Testing PPP

This section describes a different PPP setup process that allows the user to properly test the PPP connection without taking down the Ethernet interface on the Linux PC.

The demonstration of the alternate PPP setup is meant for testing and prototyping purposes only. It is not recommended to implement PPP using Linux namespaces for final production devices. If a communication path over



Ethernet is required for production devices, additional Linux networking steps (beyond those covered in this guide) may be required.

3.5.1 Create a Network Namespace

Create a Network namespace called "**ppp-testing**" on the Linux host PC, type the following command:

```
ip netns add ppp-testing
```

If desired, replace "**ppp-testing**" with any descriptive name for the new namespace.

This new namespace is completely isolated from any of the current network interfaces present on the Linux PC. To demonstrate this, issue the following command:

```
ip netns exec ppp-testing ifconfig
```

No network interfaces should appear as a result of the above command.

3.5.2 Start the PPP Session

Next, start the PPP session within the namespace with the following command:

```
ip netns exec ppp-testing pon <ppp script>
```

Replace "**<ppp script>**" with one of the filenames below:

Verizon Users:

- "vzw-SRC7611"

AT&T Users:

- "att-SRC7611"

This command will start the pppd daemon, which will configure the PPP connection, and set it as the default network connection for the namespace.

Note: *Ensure that the modem does not have an active PDP context when starting the PPP script. This will cause the PPP script to encounter an error and freeze up.*

3.5.3 Test the PPP Connection

There should now be a "**ppp0**" interface contained within the "**ppp-testing**" network namespace. To check this, issue the following command:

```
ip netns exec ppp-testing ifconfig
```

The terminal should respond with something similar to:

```
root@linux-pc:/home/debian# ip netns exec ppp-testing ifconfig
ppp0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 10.0.0.1 netmask 255.255.255.255 destination 10.0.0.1
    ppp txqueuelen 3 (Point-to-Point Protocol)
    RX packets 9 bytes 354 (354.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 393 (393.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@linux-pc:/home/debian#
```

The "ppp0" interface should be the only active network interface listed for the "ppp-testing" network namespace, as seen in the image above.

For the sake of demonstration, query the network interfaces for the Linux PC (outside of the "ppp-testing" network namespace) using the following command:

```
ifconfig
```

The "ppp0" network interface should not be listed as a network interface for the Linux PC in the response to the above command.

This is because the PPP session was run in the "container" of the Linux namespace that was created for the testing, and is not visible to anything outside of the "ppp-testing" network namespace.

To test that the PPP connection is functioning properly, issue the following command:

```
ip netns exec ppp-testing ping 8.8.8.8 -c 5
```

This command will ping Google's public DNS server in order to test the connection. The response to the ping command should look something like the image below:

```
root@linux-pc:~# ip netns exec ppp-testing ping 8.8.8.8 -c 5
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data:
64 bytes from 8.8.8.8: icmp_seq=1 ttl=55 time=159 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=55 time=117 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=55 time=155 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=55 time=155 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=55 time=152 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 117.330/148.206/159.643/15.607 ms
root@linux-pc:~#
```

If the ping command produces results similar to the image above, then it is safe to say that the Internet connection has been established.

3.5.4 Close the PPP Connection

To close the PPP connection, simply issue this command:

```
ip netns exec ppp-testing poff
```

It is a good idea to close a PPP session whenever it is no longer needed. This will prevent accidental usage of cellular data.

3.5.5 Delete the Linux Namespace

To delete the Linux namespace, issue the following command:

```
ip netns del ppp-testing
```

The Linux namespace will also be deleted upon reboot of the Linux PC, so it is not entirely necessary to delete it after testing has concluded.

3.6 PPP Troubleshooting

If the PPP scripts get hung up:

```
root@linux-pc:~# pon att-SRC7611
ATZ
OK
ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0
OK
AT+CGDCONT=1,"IP","iot0718.com.attz"
ERROR
```

1. If the Skywire has an active PDP context, the PPP scripts will freeze like this.
 - a. To fix this, open a serial or USB connection to the modem, and instruct the modem to reboot:

AT!RESET

2. The modem should be freshly powered up, or rebooted each time a PPP session is started.

If the PPP scripts timeout:

```
<ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
sent [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 0.0.0.0>
<ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
sent [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 0.0.0.0>
<ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
sent [LCP EchoReq id=0x1 magic=0xae2828]
IPCP: timeout sending Config-Requests
sent [LCP TermReq id=0x2 "No network protocols running"]
sent [LCP TermReq id=0x3 "No network protocols running"]
Connection terminated.
Modem hangup
root@linux-pc:~#
```

1. This type of error is generally caused by attempting to start the PPP session too soon after the Skywire has booted.
 - a. To fix this, wait 5-10 seconds after the modem is responsive before starting the PPP scripts.

4. QMI

4.1 System Set Up

Make sure [Section 2](#) is completed. Specifically, `libqmi-utils` needs to be installed.

Finally, make sure the SWDK is plugged into port J15 for USB use. To check, enter the following command:

```
lsusb
```

The output should be along the lines of:

```
Bus 001 Device 009: ID 1199:68c0 Sierra Wireless, Inc.
```

Note the `1199` and `68c0` from the output. If the cable is plugged into J14, the output will be similar to the following:

```
Bus 001 Device 010: ID 0403:6001 Future Technology Devices  
International, Ltd FT232 Serial (UART) IC
```

4.2 Using QMI

4.2.1 Disconnect Other Interfaces

First, disconnect from any networks, including wifi and ethernet.

Next, enter the following commands to stop the ModemManager and NetworkManager:

```
service ModemManager stop  
service NetworkManager stop
```

These commands don't produce any output to the terminal if successful.

Modem Manager and Network Manager need to be stopped so they don't try to set up the modem automatically while we are trying to. NimbeLink does not support using Modem Manager or Network Manager with the modems.

4.2.2 Gather Information

Now, enter the following command:

```
dmesg | grep cdc
```

The output should be similar to the following:

```
root@blackberrypi:~# dmesg | grep cdc  
[  9.048977] usbcore: registered new interface driver cdc_wdm  
[  9.085533] qmi_wwan 1-1.3:1.8: cdc-wdm0: USB WDM device
```

Take note of the highlighted piece of information. The yellow highlight denotes a `/dev` file and the blue highlight denotes a network interface. These will be needed later.

Next, enter the following:

```
dmesg | grep wwan
```

Output should be similar to:

```
root@blackberrypi:~# dmesg | grep wwan
[ 90.135028] qmi_wwan 3-2:1.2 wwan0: register 'qmi_wwan' at
usb-0000:00:14.0-2, WWAN/QMI device, e6:6c:5d:e5:3d:31
[ 90.135044] usbcore: registered new interface driver qmi_wwan
```

Take note of any name change for the interface, as it will be needed later. The following example assumes that this name change does not occur.

The next few commands are to check the hardware information. First, check the manufacturer:

```
qmicli -d /dev/cdc-wdm0 --dms-get-manufacturer
```

The command should respond with the following:

```
[/dev/cdc-wdm0] Device manufacturer retrieved:
Manufacturer: 'Sierra Wireless, Incorporated'
```

If it doesn't, QMI is unavailable as it is exclusive to qualcomm chipsets.

The following command confirms that the device is a SRC7611-4:

```
qmicli -d /dev/cdc-wdm0 --dms-get-model
```

Response:

```
[/dev/cdc-wdm0] Device model retrieved:
Model: 'RC7611'
```

4.2.3 Create Configuration File

Before issuing the command to start the networking, a config file needs to be created. Use any text editor, such as nano or vim, to create the file `/etc/qmi-network.conf`. Enter the following:

```
APN=<apn>
PROXY=yes
```

Replace `<apn>` with the proper APN for the chosen SIM card and cellular carrier. The proper APN for an AT&T SIM card purchased through NimbeLink is `iot0718.com.attz`. The APN for a Verizon SIM card purchased through NimbeLink is `nimblink.gw12.vzwentp`.

4.2.4 Start QMI

Once the file has been created and saved, enter the following to start QMI:

```
qmi-network /dev/cdc-wdm0 start
```

The response should be similar to the following:

```
Loading profile at /etc/qmi-network.conf...
APN: <apn>
APN user: unset
```

```

    APN password: unset
    qmi-proxy: yes
    Checking data format with 'qmicli -d /dev/cdc-wdm0 --wda-
get-data-format --device-open-proxy'...
    Device link layer protocol retrieved: raw-ip
    Getting expected data format with 'qmicli -d /dev/cdc-wdm0
--get-expected-data-format'...
    Expected link layer protocol retrieved: 802-3
    Updating kernel link layer protocol with 'qmicli -d
/dev/cdc-wdm0 --set-expected-data-format=raw-ip'...
    Kernel link layer protocol updated
    Starting network with 'qmicli -d /dev/cdc-wdm0 --wds-
start-network=apn='<apn>' --client-no-release-cid --device-
open-proxy'...
    Saving state at /tmp/qmi-network-state-cdc-wdm0... (CID:
17)
    Saving state at /tmp/qmi-network-state-cdc-wdm0... (PDH:
2267733632)
    Network started successfully

```

4.2.5 Configure Networking

Next, get the settings that QMI set up. If the command times out, wait a few seconds before trying a second time.

```
qmicli -d /dev/cdc-wdm0 --wds-get-current-settings
```

The response will be like the following:

```

[/dev/cdc-wdm0] Current settings retrieved:|
    IP Family: IPv4
    IPv4 address: 166.152.x.xxx
    IPv4 subnet mask: 255.255.255.xxx
    IPv4 gateway address: 166.152.x.xxx
    IPv4 primary DNS: 8.8.xxx.xxx
    IPv4 secondary DNS: 8.8.xxx.xxx
    MTU: 1428
    Domains: none

```

Take note of this output, as the information here will be used in the following commands to set up the networking.

The following three commands don't produce any output to the terminal, but set up the networking. For the <converted network mask>, see the table following the commands:

```

ip address add <IPv4 address>/<converted network mask> dev wwan0
ip link set wwan0 up
ip route add default via <IPv4 gateway address>

```

The IPv4 subnet mask listed above needs to be converted into slash notation. Check the following table for the conversion:

IPv4 subnet mask	/ notation
255.255.255.255	/32
255.255.255.254	/31
255.255.255.252	/30
255.255.255.248	/29
255.255.255.240	/28
255.255.255.224	/27
255.255.255.192	/26
255.255.255.128	/25
255.255.255.0	/24

After the `ip` commands have been set up and the interface is known, the network should be set. Test with the following command:

```
ping -I wwan0 -c 5 8.8.8.8
```

If the response looks like this, the session has been successfully set up:

```
ping 8.8.8.8 (8.8.8.8) from xxx.xxx.x.xxx wwan0: 56(84) bytes of data.
```

```
64 bytes from 8.8.8.8: icmp_seq=1 ttl=51 time=84.8 ms
```

```
64 bytes from 8.8.8.8: icmp_seq=2 ttl=51 time=83.0 ms
```

```
64 bytes from 8.8.8.8: icmp_seq=3 ttl=51 time=89.0 ms
```

```
64 bytes from 8.8.8.8: icmp_seq=4 ttl=51 time=68.2 ms
```

```
64 bytes from 8.8.8.8: icmp_seq=5 ttl=51 time=87.2 ms
```

```
--- 8.8.8.8 ping statistics ---
```

```
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
```

```
rtt min/avg/max/mdev = 68.233/82.468/88.970/7.410 ms
```

4.2.6 Using DNS

We can also use the Domain Name Services (DNS) from above. Open the `/etc/resolv.conf` file with any text editor (for instance, vim), and comment out or delete the existing contents (lines that start with # are commented out). Next, copy and paste the DNS lines from above into the file. The format of the file is:

```
nameserver <ip address>
```

After saving the file, ping google.com with the following command:

```
ping -I wwan0 -c 5 google.com
```

If the response is like the following, everything has been properly set up.

```
ping google.com (172.217.4.46) from 166.152.x.xxx wwan0: 56(84)
bytes of data.
64 bytes from lga15s46-in-f14.1e100.net (172.217.4.46):
icmp_seq=1 ttl=52 time=52.3 ms
64 bytes from lga15s46-in-f14.1e100.net (172.217.4.46):
icmp_seq=2 ttl=52 time=61.2 ms
64 bytes from lga15s46-in-f14.1e100.net (172.217.4.46):
icmp_seq=3 ttl=52 time=81.9 ms
64 bytes from lga15s46-in-f14.1e100.net (172.217.4.46):
icmp_seq=4 ttl=52 time=78.4 ms
64 bytes from lga15s46-in-f14.1e100.net (172.217.4.46):
icmp_seq=5 ttl=52 time=77.8 ms

--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 52.315/70.329/81.919/11.527 ms
```

4.3 Troubleshooting

If the `qmi` commands fail, make sure that `libqmi-utils` is installed, the set up procedure from [Section 2](#) has been followed, and that the `\etc\qmi-network.conf` file is named correctly and contains the correct information.

If the first few lines in the response of `qmi-network /dev/cdc-wdm0 start` says `'Profile not found'`, make sure the `/etc/qmi-network.conf` file is named correctly

5. Document Version Information

Revision	Description	Date
1	Initial release	10/21/22
2	Added NL-SW-LTE-SRC7611-4NG-B PN	03/08/24