Building a NarrowBand-IoT Base Station With USRP

CONTENTS

A Test Base Station for the Internet of Things

The term "Internet of Things" (IoT) describes the vision of connecting a vast array of things such as environmental sensors, traffic lights, city infrastructure, industrial machinery, cars, home appliances, and so on to a common network, the Internet. Interest in this is fueled by the possibility of enabling new services that could optimize processes based on the mass of aggregated data. These IoT-enabled devices and applications rely on cloud-based processing.

The terms "machine-to-machine" (M2M) and "machine-type communication" (MTC) are sometimes used in the device communication portion of the IoT to emphasize that the devices, not the actual people, are connected. The per device data traffic for this type of communication is rather low with long phases of "silence" in between transmissions. Also, the cost of M2M communication modules is expected to be low, which will enable their ubiquitous use. One example of M2M communication systems is NarrowBand-IoT (NB-IoT), which is specified and rolled out as an addition to cellular networks based on LTE.

If you're integrating IoT-enabled products, the communication function is a key supporting element in your main application. You must know whether your products can connect to the network if they're installed in a basement or a remote area, for example. You also may want to verify power consumption to guarantee battery life constraints, but you need a simple way to do this because you must focus on the main application. And you need a cost-effective test and test setup to avoid this supporting element adding to your main test burden.

You can do this with a test base station that allows you to run a simple test, such as the IMEI test. In this scenario, when you turn on the device under test (DUT), it searches for a network, connects to the test base station, and, during this process, reports its international mobile equipment ID, or IMEI, which is a number unique to the device. The test engineer verifies that the device connects successfully and reports the expected IMEI (see Figure 1). In some cases, this may already be enough to conclude that the communication module works.

Figure 1. This application waits for a DUT to connect and verifies the DUT's IMEI.

Building a Base Station for NarrowBand-IoT

To build a base station that can communicate with commercial off-the-shelf (COTS) terminal devices, you need the elements outlined in Figure 2. For the hardware, you can use the USRP (Universal Software Radio Peripheral) device to modulate and demodulate the signals exchanged with the devices at the cellular radio frequencies. The USRP device is supported by a wide variety of protocol stacks, many with roots in GNU Radio, which makes it an ideal COTS solution for a low-cost NB-IoT base station. In addition to a USRP device for the RF front end, you need a high-performance PC or laptop for real-time processing.

To operate at the frequencies of a commercial base station over the air, you must have a license from a regulatory body. In most cases, for a simple test base station application, obtaining a license is overly burdensome. To work around that, use cables to connect the USRP base station RF hardware and the terminal or, if you are missing connectors at the devices, use an RF shielded box.

NB-IOT BASE STATION TEST ARRANGEMENT

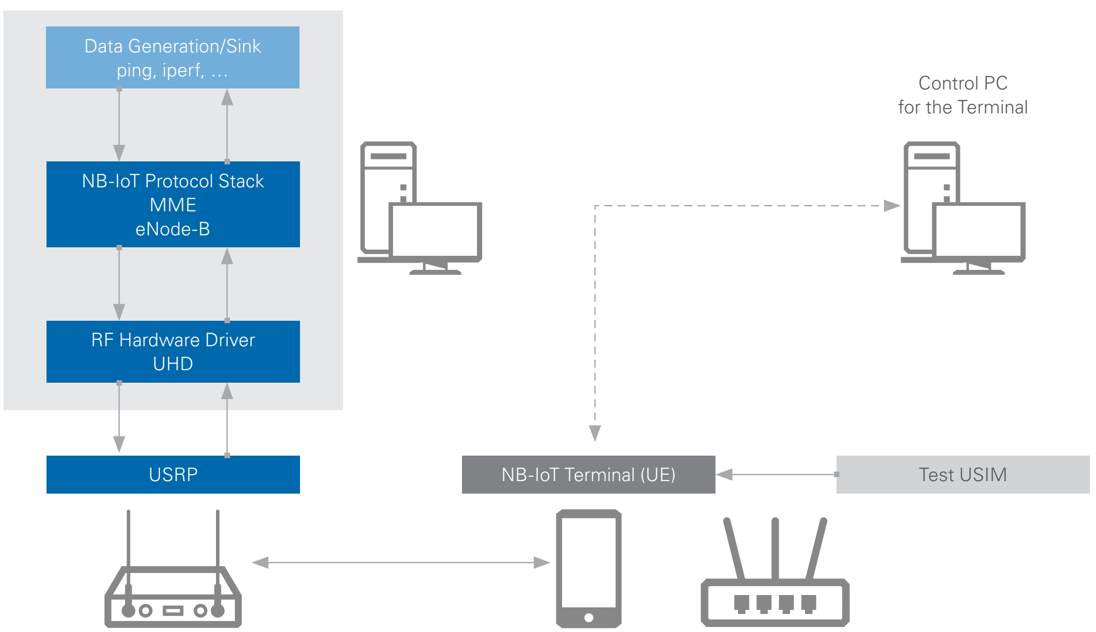

Figure 2. NB-IoT Basic Hardware and Software Arrangement

You also need processing hardware and software to implement the cellular protocol stack. In 3GPP terms, this includes the eNode-B and the mobility management entity (MME). As the network-side endpoint of the air interface with the terminal, eNode-B handles layers 1 to 3 of the protocol stack (see Figure 3). The MME is another network-side component that plays an important role in user authentication. The test application in this document requires the elements highlighted by the blue box in Figure 3.

3GPP ARCHITECTURE AND PROTOCOL STACK

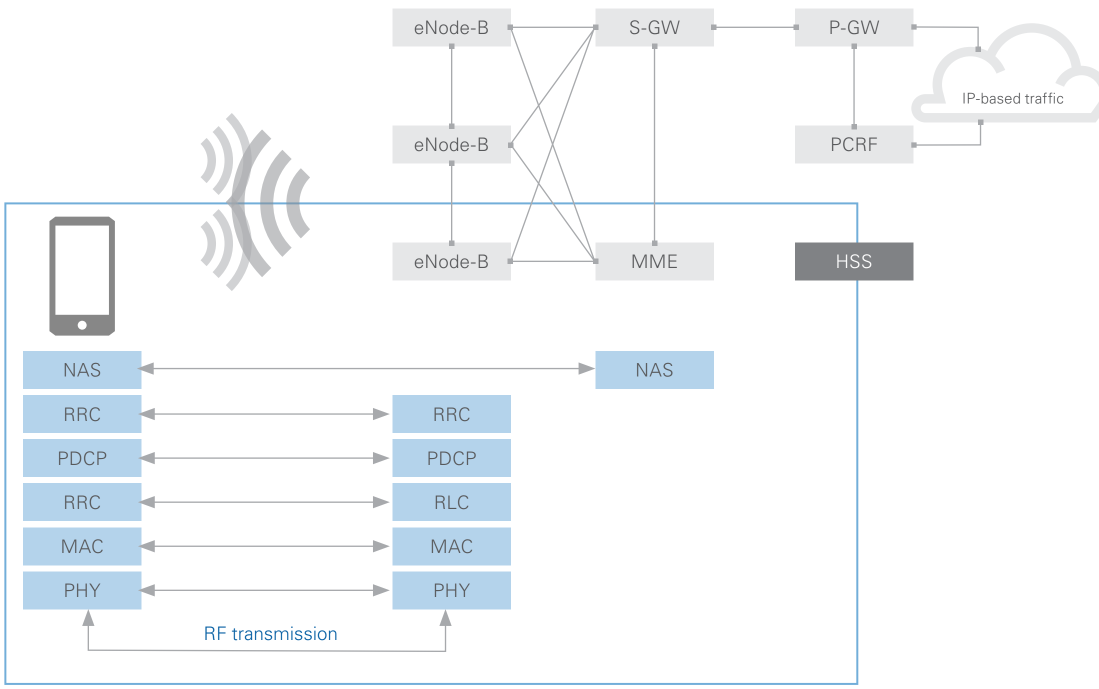

Figure 3. This 3GPP architecture and protocol stack includes layers 1 (PHY), 2 (MAC, RLC, PDCP), and 3 (RRC) of the air interface.

An important point you need to clarify early on is how the terminals—the DUTs in your application and the user equipment (UE) in 3GPP language—authenticate themselves to the (test) network. Off-the-shelf terminals that you cannot or do not want to configure in any way other than their vendor-defined normal configurations require specific identity and security parameters typically stored on an USIM card. You need to know some of that information for the MME side as well to complete the authentication successfully. Most likely, a real operator will not give you this information for an USIM card. However, for the test base station described here, you should be able to work with test USIM cards for which the security parameters are known.

Cellular protocol stacks come with varying degrees of feature completeness, run-time stability, and processing performance. If data rate is not an issue, which is the case for typical MTC applications, you can use regular, state-of-the-art PC processing hardware. Many use hardware acceleration technology such as SSE or AVX instruction sets on Intel processors. Most protocol stacks require the Linux OS. If you are not bound by other restrictions, start with the hardware and software setup your protocol vendor recommends.

This setup uses the Amarisoft LTE stack, which is commercial, closed-source software that offers performance, stability, and integration with USRP hardware. It has all the major features of an LTE and NB-IoT base station and is well documented. Users favoring open-source software can use projects such as OpenAirInterface or OpenLTE.

In this case, the Amarisoft protocols run on the regular desktop version of the Ubuntu 16.04.2 LTS Linux distribution. You also can install the low-latency kernel version, but this is not strictly required. On two different processing devices, a laptop with an Intel i7-3687U processor and a high-performance PC with an Intel i7-6822EQ CPU, the software successfully operated in real time. Two types of USRP software defined radio devices were used as the RF radio head: USRP N210 and USRP X310. The protocol stack came with configuration files for both types and for the USRP B200 series as well. For the USRP X310, both Ethernet and PCI Express connectivity between the processing device and the USRP device worked.

How You Start

Setting up the system includes (1) making all the required cable connections and (2) installing and configuring the chosen OS and software. Step 2 may require some time and effort, depending on the software suite you selected.

1. Install the Ubuntu OS.
2. Install the USRP Hardware Driver (UHD) packages (for detailed instructions, view the step-by-step guide Building and Installing the USRP Open-Source Toolchain (UHD and GNU Radio) on Linux)
3. For PCI Express communication with the USRP, install the NI USRP RIO driver stack (for detailed instructions, view the step-by-step guide Linux NI RIO Installation and Usage)
4. Test the USRP installation (see uhd_find_devices and uhd_usrp_probe in the UHD manual).
5. Install the Amarisoft LTE protocol stack.

At this point, you should start up the protocol stack and familiarize yourself with its basic usage, features, and configuration options. Amarisoft's solution comes with a command line interface, a corresponding interface for remote scripted access, and a web interface. The command line interface provides information on status and core operational parameters such as radio frequencies, cell power, and the currently attached UE devices. It also provides some tracing of L1/L2 link parameters, for example, downlink and uplink bitrates, and event counters. The web interface (see Figure 4) is good for following the sequence of messages exchanged between the base station (MME, eNB) and the UE devices. It also helps you visualize performance and statistics parameters (data rate, CPU load, and so on) and some limited options for control.

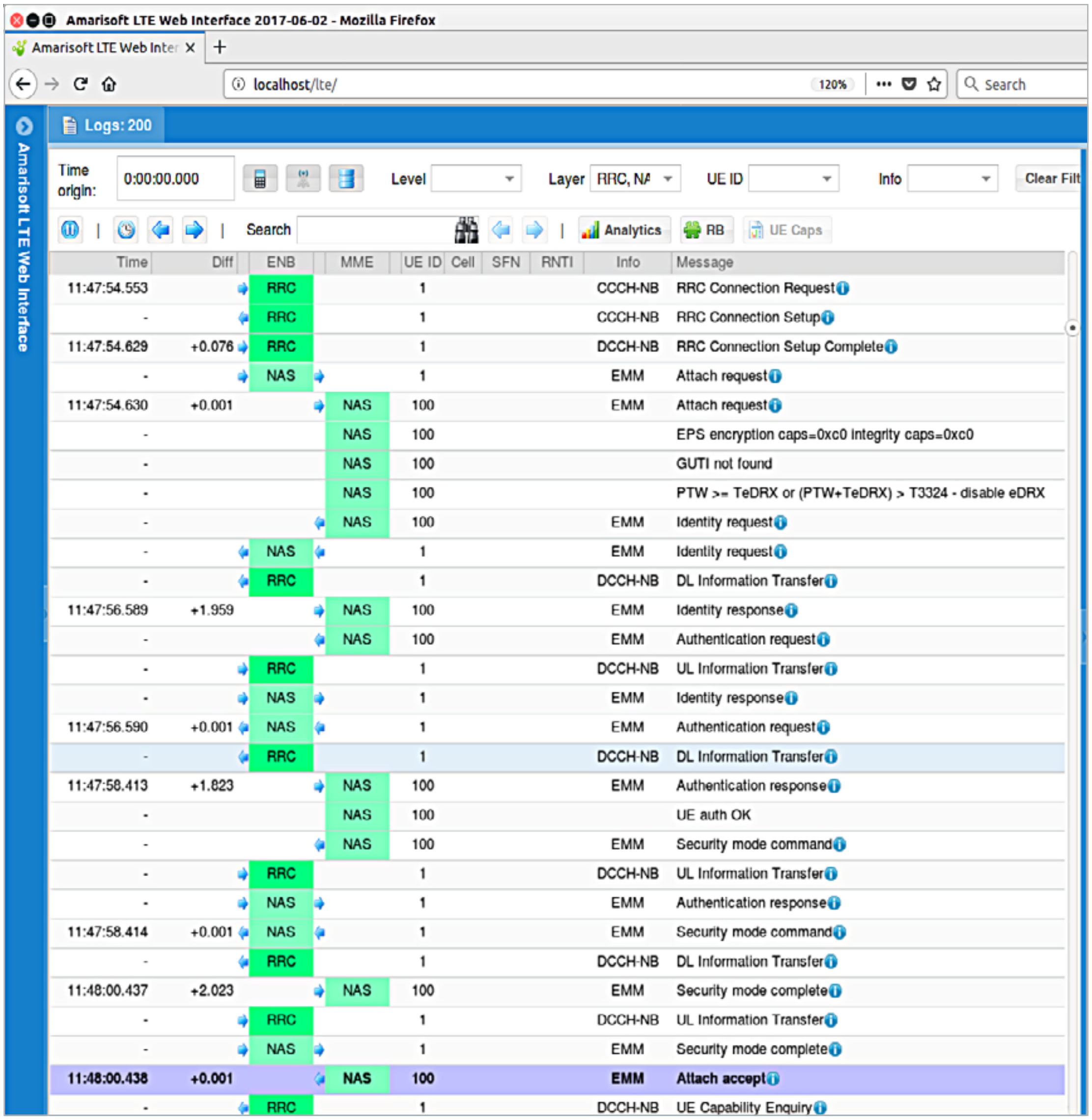

Figure 4. The base station web interface shows the exchange of messages between the base station and UE devices on any of the different protocol layers.

If possible, check on the downlink transmission that you get from the USRP device. Using a signal analyzer, you can examine the NB-IoT waveform, spectral properties, and average and peak transmission power levels. See figures 5 and 6 for an example waveform and its spectrum. If you don't have a signal analyzer, you can use a power meter to learn about average and peak transmission power. The duty cycle of the signal should be relatively low at this point because the base station transmits only synchronization and reference signals as well as occasional control messages.

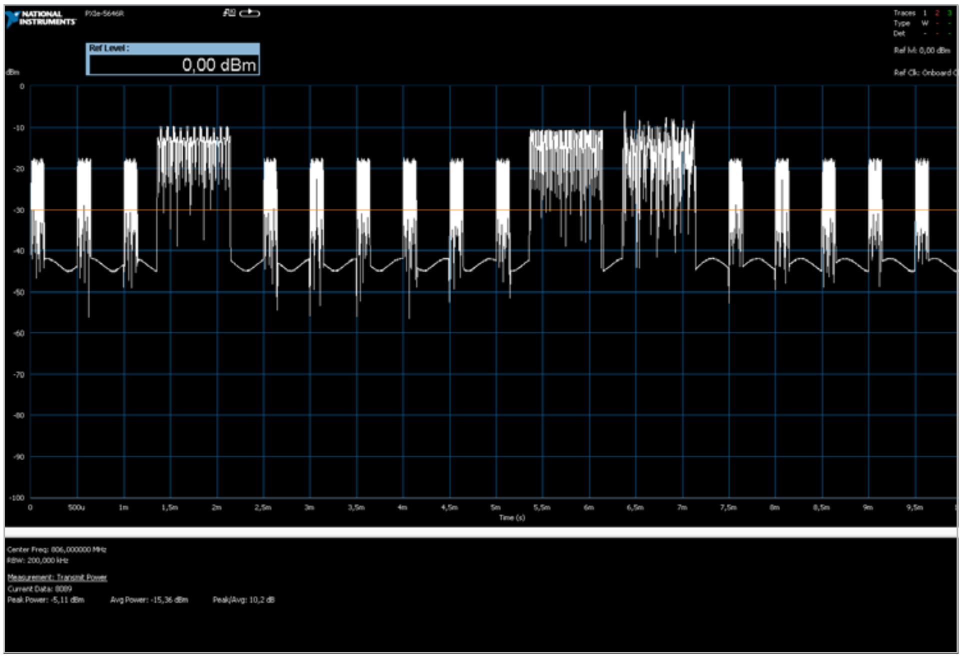

Figure 5. NB-IoT Downlink Waveform for an “Idling” Cell

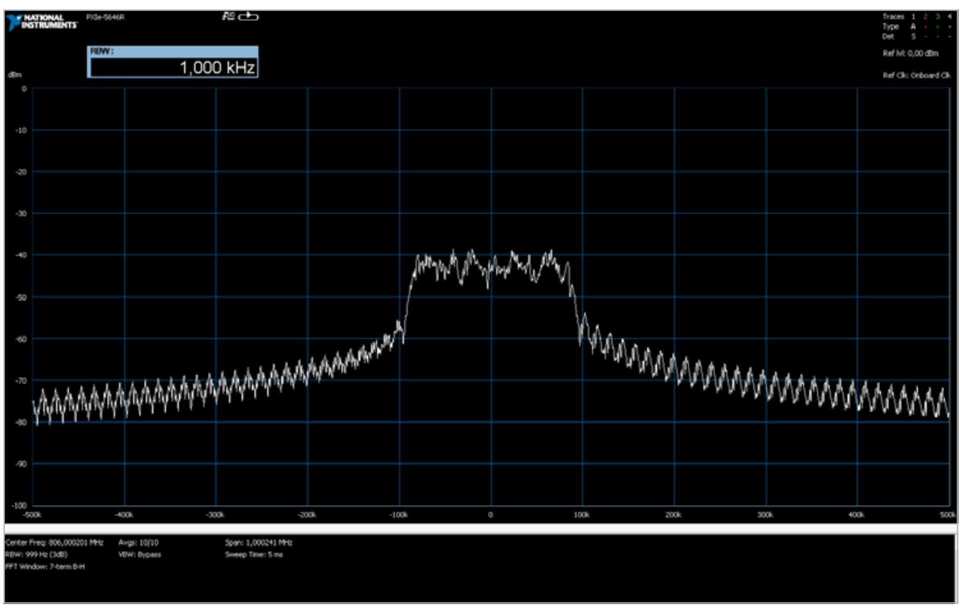

Figure 6. NB-IoT Downlink Spectrum

How you configure the NB-IoT link and the user authentication process is also important. This case is probably typical of other solutions as well; configuration files are used to capture all necessary information. For the eNB, you need to configure the basics: (1) RF center frequencies and bandwidths for downlink and uplink and (2) cell identification. For a fully operable NB-IoT link, you need to set many more parameters. These parameters compose what 3GPP defines as system information and specifies in detail in technical specification 36.331. This document also includes information on potential dependencies between certain parameters. For the MME, you should focus most on the UE authentication parameters and security algorithm configuration. The 3GPP document that specifies these is TS24.301. Note that NB-IoT is included in Release 13 and later of the 3GPP specifications.

An effective protocol stack comes with some preconfigured scenarios, and its documentation guides you through the important parameters. However, you may still need a profound understanding of the parameters and the features they control to make the test base station work with your selection of DUTs. In any case, for the first tests, start with a simple single-cell scenario that uses the most basic transmission schemes supported by both the base station and the UE. This type of scenario probably includes a stand-alone NB-IoT cell and single-tone transmission in downlink and uplink.

Making It Work

The next step is to connect an actual mobile terminal, or UE, to the base station. This scenario incorporates two Quectel IoT modules: the BC95 uses a Huawei chipset and the BG96 uses a Qualcomm chip.

As mentioned earlier, you should make sure you don't interfere with any real-world licensed applications such as a life network. If possible, use appropriate RF cables to connect the UE and base station. This has the added benefit of reproducible conditions and results.

Before you turn on anything, determine the required path attenuation between the base station and the UE. For that, you need to know the transmission power and the range for receive power levels (see Table 1). Consult the USRP and UE device specifications for exact numbers. Ensure that the attenuation you choose keeps the received power well within the allowed range even with maximum transmission power. If the attenuation is too low, you risk damaging the receiver. If the attenuation is too high, you degrade the signal quality too much for the receiver to decode the transmitted message. This example uses 50 dB of attenuation at a maximum base station output power in the range of -10 dBm to -5 dBm.

	Base station	UE
Transmission power	Measured peak power or maximum output power of the USRP device	Up to 23 dBm (UE power class 2)
Receive power range	Sensitivity … RX input damage power level	

Table 1. Transmission and Receive Power Levels for the Base Station and the UE

For this task, you should know how to make the UE search for the network and connect to it. You may have to configure the UE for the frequency band or even the carrier frequency (or a frequency index termed EARFCN by 3GPP). The UE may do everything else automatically once you turn it on. If you need to trigger the individual procedures yourself, the following should happen in the simplest case:

- The UE searches the frequency ranges (bands) it supports for available networks. With only the test base station transmitting (over cable), the UE should find a single cell with the cell identification that the base station is configured with.
- The UE decides to connect to the network using the found cell. In 3GPP vocabulary, it initiates an "attach" procedure. The result of successfully attaching to the network is a route between the UE and the network. The MME and the eNode-B know the UE, and the MME assigns the UE an IP address. After that, regular data communication is possible.

The attachment procedure consists of several subprocedures that take advantage of the entire protocol stack's wide range of capabilities. First, the UE sets up a connection with the base station on the radio resource control (RRC) layer; the procedure is called RRC connection setup. It begins by transmitting a series of so-called random access preambles with increasing transmission power until the base station addresses a message called random access response to the UE. Then the UE sends more information about itself in what is known as "Message 3." This sends more higher and lower layer messages back and forth. These may include further random access procedures to set up security and other items before, finally, the network sends an ATTACH ACCEPT message. Figure 7 shows a conceptual message sequence chart for the attachment procedure. Figures 8 and 9 show how eNode-B and MME record an attacnhment procedure.

3GPP ATTACHMENT PROCEDURE

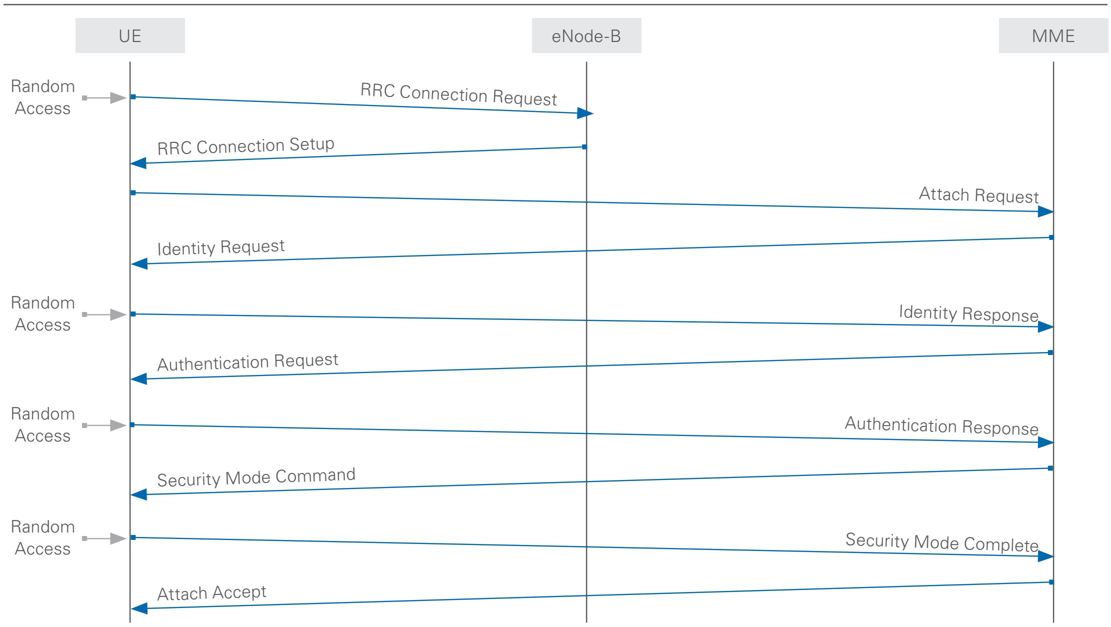

Figure 7. Higher Layer View of the Message Flow Between the UE and the Network During a UE Attach Procedure

```
root@ni-desktop: ~/mme
(enb) UHD: Loaded /root/.uhd/cal/tx_iq_cal_v0.2_F4F5DF.csv
UHD: Loaded /root/.uhd/cal/tx_dc_cal_v0.2_F4F5DF.csv
UHD: Loaded /root/.uhd/cal/rx_iq_cal_v0.2_F4F5DF.csv
Chan Gain(dB)  Freq(MHz)
 TX1      0.0 806.000000
 RX1     20.0 847.000000
t g
Press [return] to stop the trace
--#UE--------       --RRC----------- --DL------------ --UL------------
conn idle disc prach reqst reest pagng  retx  txok brate  retx  rxok brate
   0    0    0     0     0     0     0     0     0     0     0     0     0
   0    0    0     0     0     0     0     0     0     0     0     0     0
   0    0    0     0     0     0     0     0     0     0     0     0     0
   0    0    0     0     0     0     0     0     0     0     0     0     0
   0    0    0     0     0     0     0     0     0     0     0     0     0
   0    0    0     0     0     0     0     0     0     0     0     0     0
   1    0    0     2     0     0     0     0     0     0     0     0     0
   1    0    0     1     0     0     0     0     4   356     0     4   500
   1    0    0     1     0     0     0     0     1   112     0     2   208
   1    0    0     1     0     0     0     0     2   412     0     2   208
--#UE--------       --RRC----------- --DL------------ --UL------------
conn idle disc prach reqst reest pagng  retx  txok brate  retx  rxok brate
   2    0    0     2     0     0     0     0     1   252     0     2   208
   1    0    0     0     0     0     0     0     0     0     0     2   116
   1    0    0     0     0     0     0     0     0     0     0     0     0
   1    0    0     0     0     0     0     0     0     0     0     0     0
   1    0    0     0     0     0     0     0     0     0     0     0     0
   1    0    0     0     0     0     0     0     0     0     0     0     0
   1    0    0     0     0     0     0     0     0     0     0     0     0
   1    0    0     0     0     0     0     0     0     0     0     0     0
   1    0    0     0     0     0     0     0     0     0     0     0     0
   1    0    0     0     0     0     0     0     0     0     0     0     0
--#UE--------       --RRC----------- --DL------------ --UL------------
conn idle disc prach reqst reest pagng  retx  txok brate  retx  rxok brate
   1    0    0     0     0     0     0     0     0     0     0     0     0
```

Figure 8. The base station traces show the UE connecting to the base station. The first column is the connection counter, the "prach" column records detected random access attempts, and the DL and UL "brate" (bitrate) columns show corresponding (control) data exchange.

```
root@ni-desktop: ~/mme
root@ni-desktop:~/mme# ./ltemme config/mme-ni.cfg
LTE MME version 2017-04-28, Copyright (C) 2012-2017 Amarisoft
This software is licensed to National Instruments.
(mme) ue
            IMSI            IMEISV      M_TMSI REG          TAC #ERAB IP_ADDR
 001010123456789 8637030301230000 0x55cd052d   Y  00101.   0x2     1 192.168.3.2
(mme)
```

Figure 9. After a successful attach procedure, the MME knows the UE (its IMSI and other parameters) and assigns it an IP address.

Successfully attaching the UE is no small achievement. Don't be discouraged if your first attempts don't get you that far. Debugging can be complex and time consuming, but Table 2 offers a few guidelines that may make the process easier. Tools that can help you include logging information and log viewing software for the base station protocol stack, the UE and signal analyzers to peek into the actual RF waveforms, and measurement equipment such as a power meter and signal analyzer.

<table>
<tr><th>No.</th><th>Symptom</th><th colspan="3">What to do</th></tr>
<tr><td rowspan="3">1</td><td rowspan="3">UE cannot detect a cell</td><td colspan="3">Ensure that the UE is configured for the band you’re using with the base station (or vice versa)</td></tr>
<tr><td colspan="3">Check on base station transmit power and UE receive power levels; try varying the power (attenuation between base station and UE) because the received power may be too high (saturating the receiver) or too low (below sensitivity threshold)</td></tr>
<tr><td colspan="3">Vary the physical layer cell ID</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Base station sees no UE activity on any layer, but UE detected the correct cell</td><td colspan="3">Use a signal analyzer (for example, max hold spectrum) to detect any RF activity on the uplink (see notes on next page)</td></tr>
<tr><td colspan="2">No activity:
Does the UE need specific configurations or triggers to start transmission?
Check random access timing configuration (see 2a)</td><td>Activity:
Check power versus time over a longer period; trigger off the first uplink activity (power threshold) to observe UE power ramping behavior (go to 2b or 2c)</td></tr>
<tr><td>2a</td><td>UE does not transmit anything</td><td colspan="3">Verify that the random access configuration allows for enough transmission occasions; the SIB2 parameter Nprach-Periodicity-r13 should be one of the values specified by the 3GPP; try varying the value</td></tr>
<tr><td>2b</td><td>UE transmits multiple times, each transmission with same power</td><td colspan="3">Verify the power ramping parameters for the preamble transmissions (see 2c, power ramping)</td></tr>
<tr><td rowspan="2">2c</td><td rowspan="2">UE transmits with increasing power and then stops, potentially repeating after some time</td><td colspan="3">The base station is not able to detect an incoming random access preamble; potential causes include the following:</td></tr>
<tr><td colspan="2">Even after undergoing full power ramp, the preamble power is too low to be received</td><td>Wrong timing of preamble transmission or reception</td></tr>
<tr><td rowspan="2"></td><td>Power ramping</td><td colspan="3">Verify the values of the SIB2 powerRampingParameters-r13; the UE starts transmitting with a power of preambleInitialReceivedTargetPower plus a correction for the pathloss it estimated using the downlink signal; for each subsequent preamble transmission (within the same random access procedure), it increases power by powerRampingStep</td></tr>
<tr><td>Timing issue</td><td colspan="3"></td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">Base station detects a random access preamble and sends a random access response (RAR), but it does not receive Message 3</td><td colspan="3">Use a signal analyzer to check power versus time over a longer period (see notes on next page); after the preamble power ramping, there should be a transmission that lasts a little longer and has more amplitude variation (Message 3) than the preamble transmissions</td></tr>
<tr><td colspan="2">No Message 3 transmission</td><td>Message 3 transmitted but not decoded by base station</td></tr>
<tr><td rowspan="6"></td><td rowspan="4">No Message 3 transmitted</td><td colspan="3">If possible, check the following (for example, from base station and UE logs):</td></tr>
<tr><td colspan="2">The preamble ID that the UE used for preamble transmission equals what the base station detected</td><td>The UE received the RAR and determined that it was addressed to the preamble ID, so the UE chose to transmit the preamble</td></tr>
<tr><td colspan="3">Perhaps the UE did not receive the RAR because of unfavorable configuration. In this case, the following combinations of values worked well (parameters included in the system information):</td></tr>
<tr><td colspan="3"><table>
<tr><th rowspan="5">npdcch-Start
SF-CSS-RA-r13</th><th></th><th colspan="4">npdcch-Offset-RA-r13</th></tr>
<tr><th></th><th>0</th><th>oneEighth</th><th>oneFourth</th><th>threeEighths</th></tr>
<tr><td>v2</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td></tr>
<tr><td>v4</td><td>✓</td><td>✓</td><td>✓</td><td>✓</td></tr>
<tr><td>v8</td><td>✓</td><td>✓</td><td>(✓)</td><td></td></tr>
</table></td></tr>
<tr><td rowspan="2">Message 3 transmitted</td><td colspan="3">Bad signal quality at the base station side that’s possibly due to low SNR or impairments</td></tr>
<tr><td>Try increasing Message 3 power (preambleInitialReceivedTar-getPower in the powerRamp-ingParameters of the system information)</td><td>Run an IQ image and DC offset calibration for your radio unit</td><td>If possible with the protocol stack and radio unit, consider applying an LO offset to move the used portion of the spectrum in the baseband away from DC</td></tr>
<tr><td rowspan="2">4</td><td rowspan="2">Attach rejected, specifically PDN connectivity rejected, or attach accepted but UE lacks IP address</td><td colspan="3">The UE may not be able to transmit data over data radio bearers but uses the control plane IoT (CIoT) optimization method</td></tr>
<tr><td colspan="3">Enable CIoT optimization in the MME</td></tr>
<tr><td>5</td><td>Many retransmissions on the downlink and/or uplink data channels, excessive random access procedures, or even frequent connection losses</td><td colspan="3">Look for UHD warnings indicating Tx buffer underruns or Rx buffer overflows. These happen if the protocol stack cannot provide DL baseband data or retrieve UL baseband data fast enough, which leads to data loss. If this happens a lot, the likelihood of affecting the data exchange between the base station and UE increases. The protocol stack can recover using automatic retransmission mechanisms, but you may want to optimize UHD settings such as send and receive frame sizes as well as Tx and Rx bitrates.</td></tr>
</table>

Table 2. Tips for Debugging the Connection Setup

Notes:

Use a signal analyzer to check power versus time over a longer period:

- Trigger off the first uplink activity (power threshold)
- Configure a low measurement bandwidth (NB-IoT signals have low bandwidth; if operated in-band with LTE, the larger LTE bandwidth may apply)
- Configure a relatively long measurement interval, some seconds

Selected SIB1 and SIB2 Parameters Used

SIB2

```
powerRampingParameters-r13 {
                powerRampingStep dB6,
                preambleInitialReceivedTargetPower dBm-90,
              },
nprach-ParametersList-r13 {
  {
      Nprach-Periodicity-r13 ms320,
      ...
      npdcch-StartSF-CSS-RA-r13 v2,
      npdcch-Offset-RA-r13 oneEighth,
   }
}
      npdsch-ConfigCommon-r13 {
              nrs-Power-r13 0
            },
uplinkPowerControlCommon-r13 {
              p0-NominalNPUSCH-r13 -20,
              alpha-r13 al09,
              deltaPreambleMsg3-r13 6,
            }
```

SIB1

```
p_max_enable: true,
p_max: 20,          /* SIB1.p-Max */

msg3_n_rep: 4,
npusch_single_tone_i_tbs: 10,
npusch_n_rep: 2,

inactivity_timer: 100000,
```

End Product

Once you have sorted out the interoperability issues, the IoT-enabled terminal attaches to the test network and you can explore in more detail the communication aspects relevant to your application. Figure 10 shows a message sequence chart from the base station log and highlights the "attach accept" message. Figure 11 is the UE view after a successful attach procedure. First, you may want to try sending a ping between the base station and the UE. Note that for NB-IoT, targets for low data rates and low duty cycle per device ping times may appear a bit large (see Figure 12).

You now have a running connection between the test base station and the mobile device! From here, you can start building your test application, develop test cases that check user experience or dig deeper to verify physical layer performance, or simply demonstrate your IoT device.

For more help building a USRP NB-IoT solution, use the Ettus USRP product selector to help you choose the FPGA, RF daughtercard, and form factor that fit your requirements.

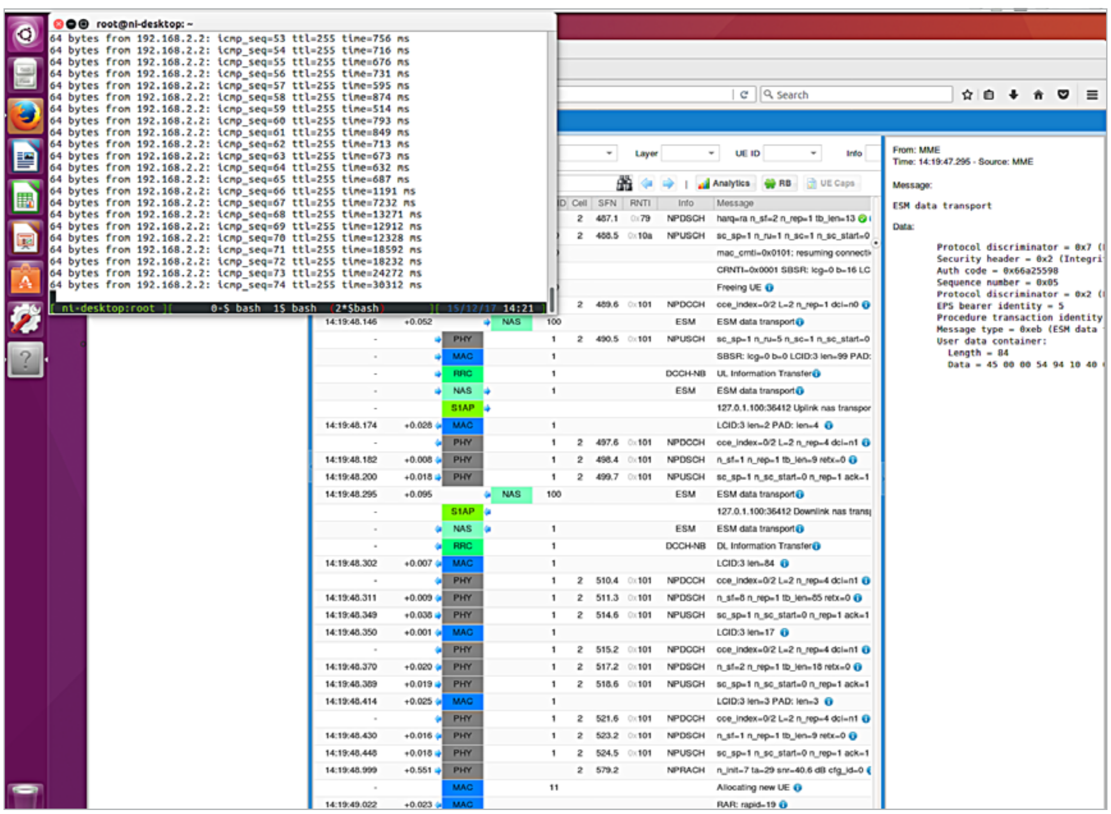

Figure 10. This detailed logging of exchanged messages at the base station side shows that the base station has sent the attach accept message to the UE.

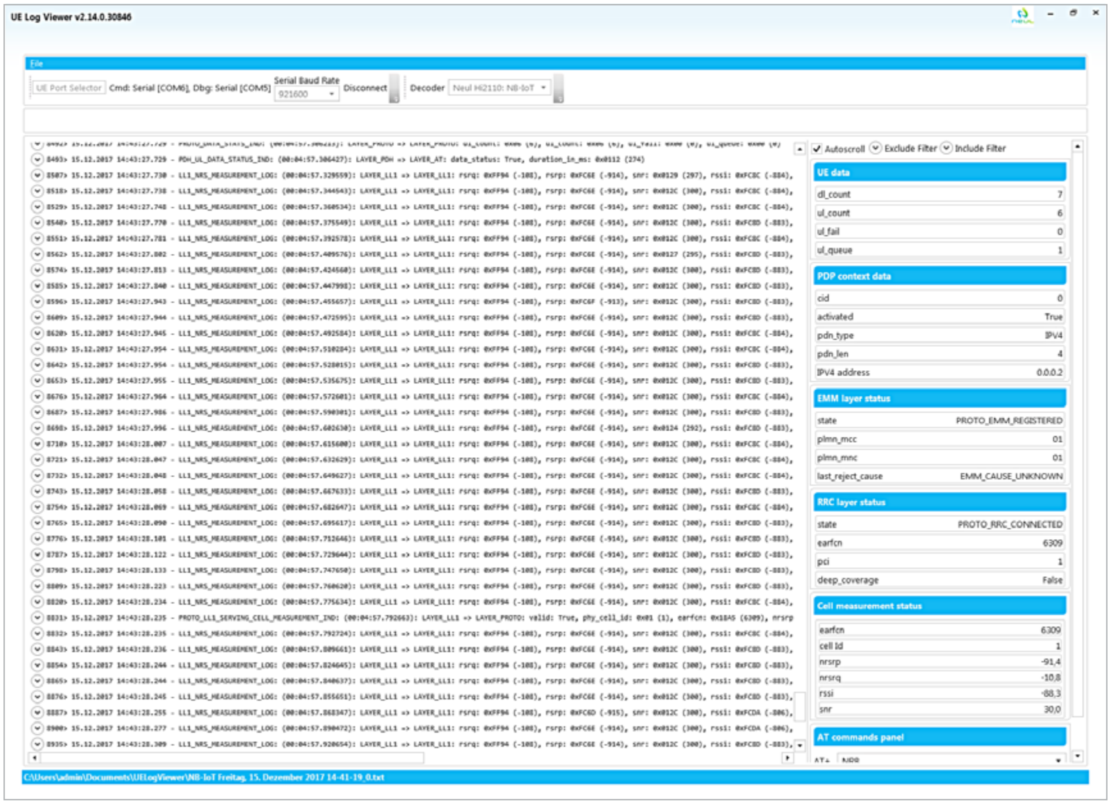

Figure 11. This UE view after attaching to the network shows that the UE is connected and registered.

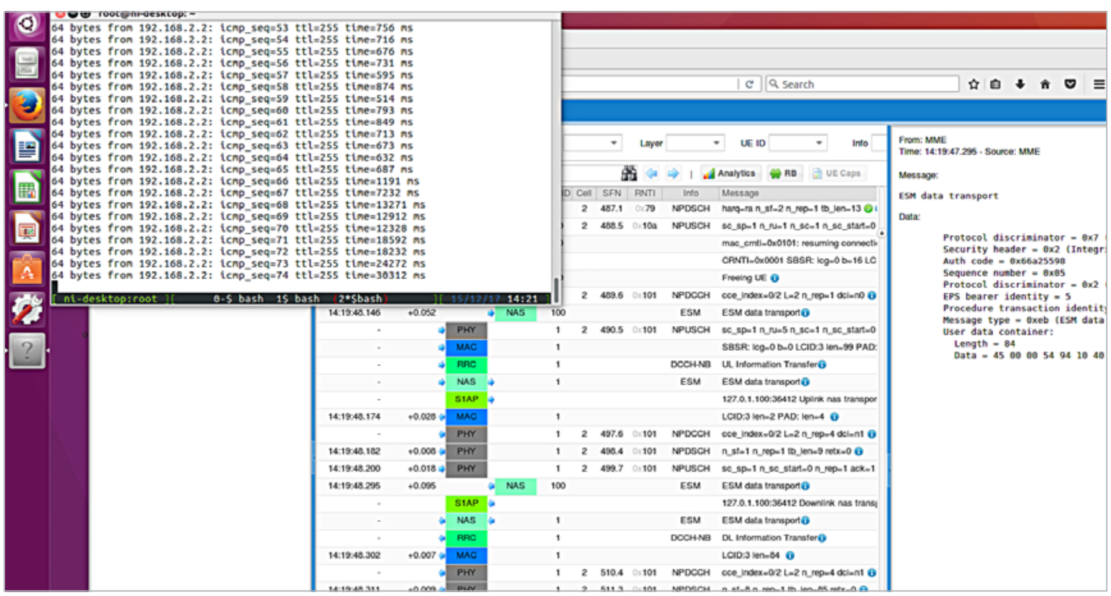

Figure 12. The UE is pinged from the network side.