LTE MIMO Application Example 2x2 Demo Script

Demonstrates how the LTE Application Framework 2.0.1 can be extended to MIMO (2x2).

Requirements

Software requirements

- Microsoft Windows 8.1/7 (64-bit) preinstalled on the PXI controller
- NI-USRP 15.5
- LabVIEW Communication System Design Suite 2.0
- LTE Application Framework 2.0.1
- Extract zip file "LTE MIMO2x2 Extensions v2_0_1.zip"

The installation of NI software is started by running `setup.exe` from the provided installation media. Follow the installer prompts to complete the installation process.

Hardware requirements

The following hardware is required.

- Chassis: Recommended NI PXIe-1085 with 2 MXI-Adapters; a second chassis is needed when using different host computers.
    - PXI chassis should have 16 GB RAM as 2 application frameworks are running in parallel.
- Controller: Recommended NI PXIe-8135; a second controller is needed for a separate host operation.
- 2 USRPs: NI USRP-294xR/295xR with 40 MHz, 120 MHz, or 160 MHz
- Cabled setup:
    - 2 SMA cables: Female/female cable, included with the NI USRP RIO device
    - 2 Attenuators with 30 dB attenuation
- Over-the-Air (OTA) setup:
    - 4 Antennas: Refer to *Using Over-the-Air Transmission* for more information about this mode.[1]

For 294xR USRPs, an external clock source is required. The eNodeB USRP can act as a master for the user equipment (UE) USRP.

The preceding recommendations assume you are using PXI-based host systems. Alternatively, a PC with two PCI-based or PCI Express-based MXI adapter can be used. Make sure your host has at least 20 GB of free disk space and 16 GB of RAM. In case of using two Labtops with an Express card-based MXI adapter, 8 GB RAM for each would be sufficient. To compile bitfiles for the NI RF device FPGA, your system should be equipped with 16 GB RAM.

[1] Consider local laws if you are transmitting over-the-air. The USRP-29xx is not approved or licensed for transmission over-the-air using an antenna.

RF Cabling

To configure the USRP RIO setup using RF cabling:

1. Ensure both NI USRP devices are properly connected to the host system/systems running LabVIEW. Use MAX, or **System Diagram** in LabVIEW Communications, to check or change the hardware aliases.
2. Create an RF connection as shown in Figure 1.
3. On station A acting as eNodeB, connect two cables to RF0/TX1 and RF1/TX1.
4. Connect the other ends of those two cables to 30 dB attenuators.
5. On station B acting as UE, connect the other ends of the attenuators to RF0/RX2 and RF1/RX2.
6. Power on the USRP devices.
7. Power on the host system/systems.

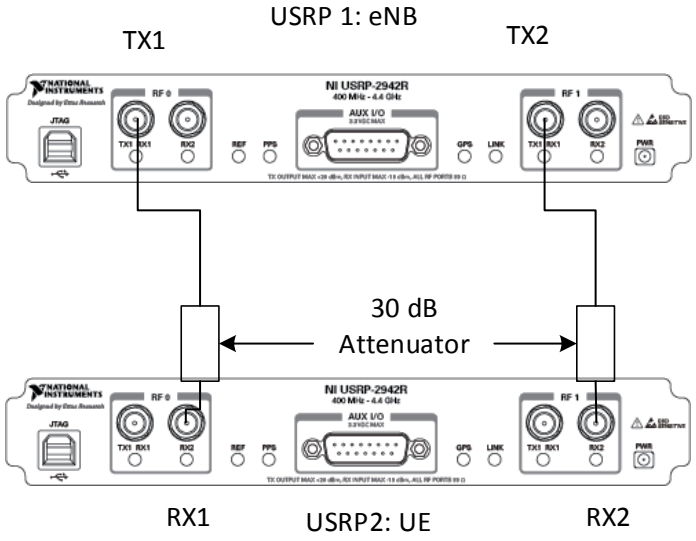

Figure 1: RF cabling

OTA Connection

To configure the USRP RIO setup using the antennas:

1. Ensure both NI USRP devices are properly connected to the host system/systems running LabVIEW. Use MAX, or **System Diagram** in LabVIEW Communications, to check or change the hardware aliases.
2. Create an RF connection as shown in Figure 2.
3. On station A acting as eNodeB, connect two antennas to RF0/TX1 and RF1/TX1.
4. On station B acting as UE, connect two antennas to RF0/RX2 and RF1/RX2.
5. Power on the USRP devices.
6. Power on the host system/systems.

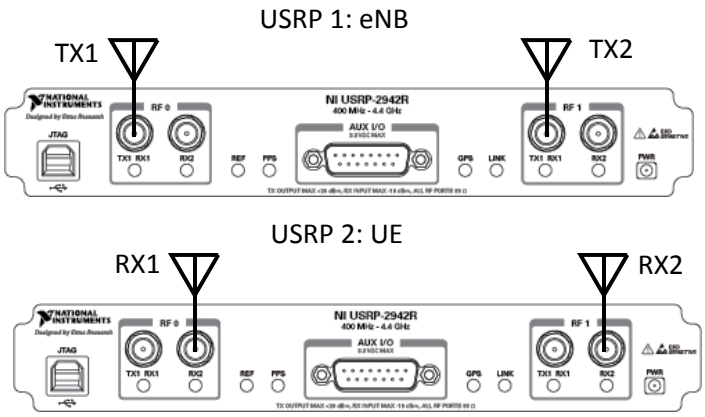

Figure 2: OTA Connection

System Structure

Figure 3 shows the block diagram of the eNodeB and the UE in the downlink (DL).

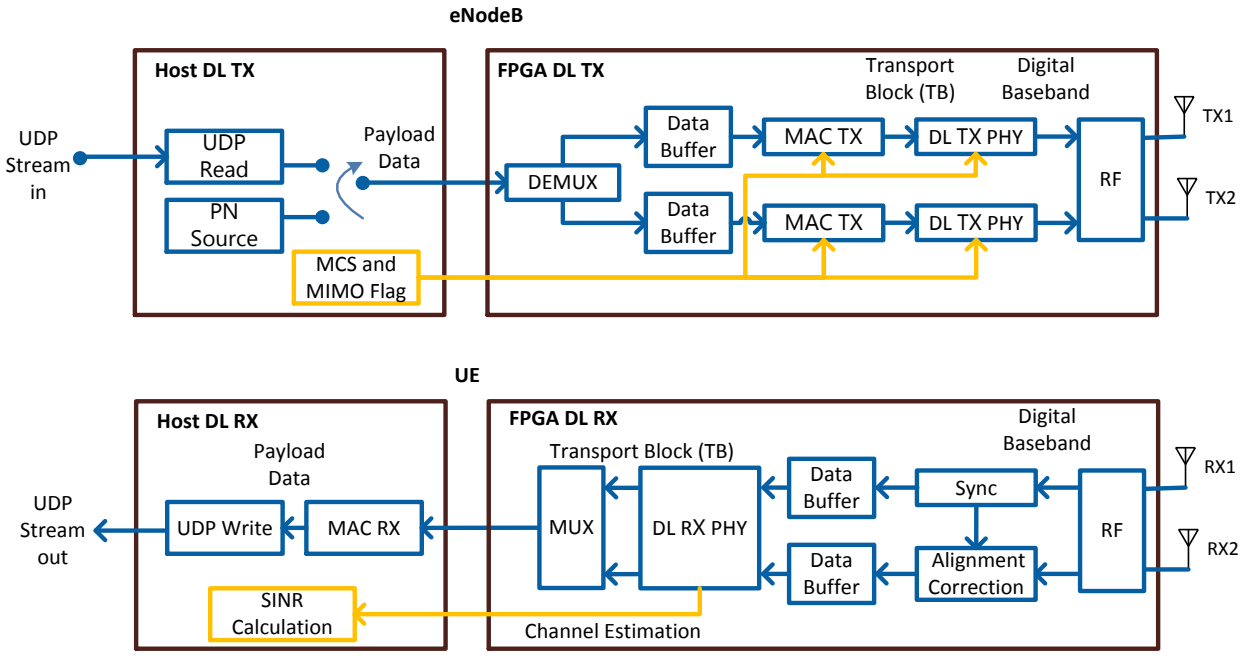

Figure 3: Block diagram of the system

The components shown in Figure 3 perform the following tasks:

- **UDP read:** Reads data, provided by an external application, from a UDP socket. The data is used as payload data in the transport block (TB). This data is then encoded and modulated as an LTE DL signal by the downlink transmitter (DL TX PHY).
- **UDP write:** Writes the payload data, which was received and decoded from the LTE DL signal by the downlink receiver (DL RX PHY), to a UDP socket. The data can then be read by an external application.

- **MCS and MIMO Flag:** Creates a cluster including the required information for the downlink control information (DCI) message such as the modulation and coding scheme (MCS) and the MIMO flag. The MIMO flag is a control that is used to activate MIMO or SISO in the eNodeB.
- **DEMUX:** A demultiplexer is used to split the data stream to Data Stream 1 and Data Stream 2 if the transmitter is configured to use both RF chains for TX.
- **Data Buffer:** Two buffers are used to store data coming from the Payload H2T FIFO into two local FIFOs for TX chain 1 and TX chain 2. In the RX mode, two buffers are used to store the received data coming from the synchronization module to synchronize the signal processing in RX chain1 and RX chain 2.
- **MAC TX:** A simple MAC implementation that adds a header to the TB containing the number of payload bytes. The header is followed by the payload bytes, and the remaining bits of the TB are filled with padding bits.
- **DL TX PHY:** Physical layer (PHY) of the downlink (DL) transmitter (TX). Encodes the physical channels and creates the LTE downlink signal as digital baseband I/Q data. This code includes encoding of the control channel (PDCCH), encoding of the data channel (PDSCH), resource mapping, and orthogonal frequency-division multiplexing (OFDM) modulation. Look to the **white paper** of LTE Application Framework 2.0.1 to get more information about TX Bit Processing and IQ Signal Processing.
- **Sync:** The primary synchronization sequence (PSS) is used for synchronization. The synchronization is done using the received signal on RF0/RX2. The synchronization results such as frame alignment and frequency offsets estimation (integral and fractional) are used to adjust the received signals on both ports RF0/RX2 and RF1/RX2.
- **DL RX PHY:** Physical layer (PHY) of the downlink (DL) receiver (RX). Demodulates the LTE downlink signal and decodes the physical channels, OFDM demodulation, resource demapping, MIMO channel estimation and MIMO equalization, decoding of the control channel (PDCCH), and decoding of the data channel (=shared channel, PDSCH). For MIMO equalization, three algorithms have been implemented namely: simple algorithm, minimum mean squared error (MMSE), and zero-forcing (ZF). The simple algorithm means that the inverse of the estimated channel parameters is used directly in the equalizer.
- **MUX:** A multiplexer is used to combine the received data from both RX chains to be transmitted to the host.
- **MAC RX:** Disassembles the TB and extracts the payload bytes.
- **SINR calculation:** Calculates the signal-to-interference-noise-ratio (SINR) based on the channel estimation that was used for PDSCH decoding. Channel estimation is either based on cell-specific reference signals (CRS), or on UE-specific reference signals (UERS).
- **RF:** In the transmit path, it performs digital up conversion (DUC), RF impairments correction, and writes the transmit data to the RF. In the receive path, it reads the received data from the RF, performs digital down conversion (DDC) and performs RF impairments correction.

Running the LabVIEW Host Code

Caution Ensure you meet the hardware and software requirements before trying to run the host code.

To use the LTE MIMO Application Example:

1. Launch LabVIEW Communications System Design Suite 2.0 by selecting **LabVIEW Communications 2.0** from the Start menu.
2. From the Project Templates on the launched **Project** tab, select **Application Frameworks,** then select **LTE Design USRP RIO v2.0.1** to launch the project.

3. Save the created project in an appropriate folder.
4. Copy the extracted zip file "**LTE_MIMO2x2_Extensions v2.0.1**" to the previous project folder and include it to the project.
5. The "**LTE_MIMO2x2_Extensions**" has a number of SubVIs, and the new FPGA design with the corresponding bitfile. The following files and folders should be included to the main project:
   a. **LTE AFW Resources MIMO.grsc**
      The required resources for the MIMO setup.
   b. LTE Host DL MIMO.gvi
      This top-level host VI implements a downlink transmitter (eNodeB) and a downlink receiver (UE).
   c. **MIMO**
      This folder contains host and FPGA subVIs, which were specifically designed for the LTE MIMO Application Example. It has four subfolders: **FPGA DL TX**, **FPGA DL RX**, **FPGA Shared**, and **Host**. The FPGA subfolders should be included as "USRP" while the host subfolder should be included as "PC" to the main project.
   d. **USRP RIO**
      This folder contains the Top-level FPGA VI "**LTE FPGA USRP RIO DL MIMO.gvi**". In addition, it has two subfolders: **FPGA** and **Host**. They contain the required subVIs for the MIMO setup. They should be included such as the previous.
   e. Builds
      In this folder, three bitfiles are created based on the USRP RIO bandwidth:

- **LTE FPGA USRP RIO 40 MHz BW DL MIMO2x2.lvbitx**: Used for USRO RIO devices with 40 MHz bandwidth.
- **LTE FPGA USRP RIO 120 MHz BW DL MIMO2x2.lvbitx**: Used for USRO RIO devices with 120 MHz bandwidth.
- **LTE FPGA USRP RIO 160 MHz BW DL MIMO2x2.lvbitx**: Used for USRO RIO devices with 160 MHz bandwidth.

6. Save the LTE instance project after including the MIMO extensions.
7. The eNodeB and UE should be running using two instances of the above LabView project. If the eNodeB and UE are running using a single project, some nodes will be used twice in eNodeB and in UE. The performance will be degraded. Therefore, the above project (the LTE instance project with the MIMO extension) should be copied to another folder.
8. If two host systems are used, copy the saved project to the another host. Two projects are available, once will act as eNodeB and once will act as UE.

On the eNodeB

1. Open the project "**LTE Design USRP RIO v2.0.1.lvproject**" from the first folder.
2. Open the host top-level "**LTE Host DL MIMO.gvi"** located in **MIMO Extensions** folder. The front panel of this VI is shown in Figure 4. The description of all controls and indicators can be found in the **Getting Started Guide of LTE Application Framework** 2.0.1.
3. Configure the RIO identifier in the **RIO Device** control. Use MAX to configure the hardware alias of the USRP RIO that you want to use as eNodeB.
4. Select the USRP bandwidth.
5. Run "**LTE Host DL MIMO.gvi"** by clicking the run button ( ).
   - If successful, the **FPGA Ready** indicator lights.

- If you receive an error, check if your RIO device is connected properly.

6. Set **eNB TX Frequency [Hz]** to a frequency supported by your NI USRP RIO.
7. Enable the **eNB Transmitter,** which implements a Downlink Transmitter, by setting the switch control to **On.**
    - If successful, the **eNB TX Active** indicator lights.
8. Enable the downlink MIMO by setting the switch control **MIMO flag** presented in Figure 4 to **On.**

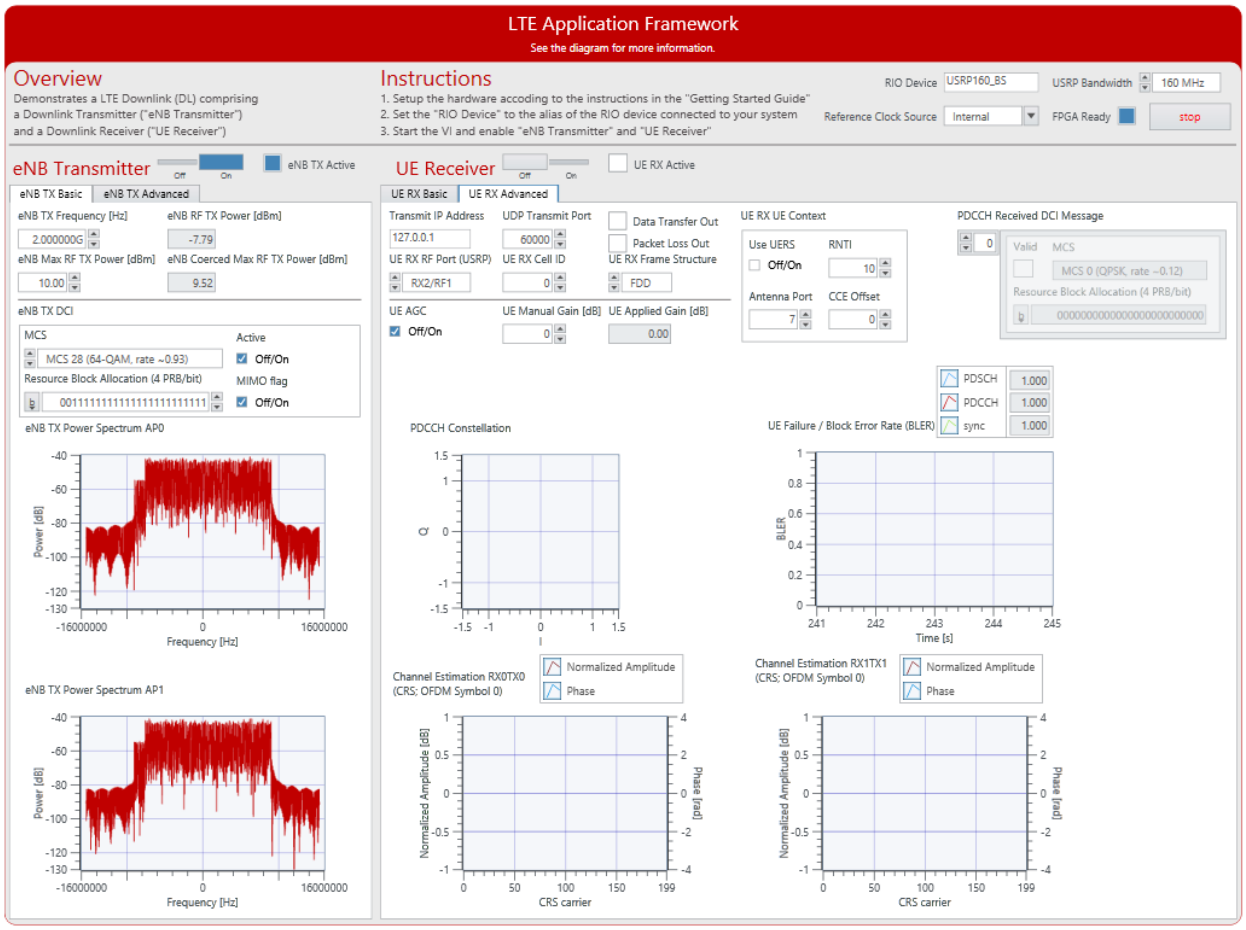

Figure 4: Front panel of eNodeB

On the UE

1. Open the project "**LTE Design USRP RIO v2.0.1.lvproject**" located in the second folder.
2. Open the host top-level "**LTE Host DL MIMO.gvi"** located in **MIMO Extensions** folder. The front panels of this VI is shown in Figure 5 and Figure 6. The description of all controls and indicators can be found in the **Getting Started Guide of LTE Application Framework 2.0.1**. The extra controls and indicators for MIMO are described in the following section.
3. Configure the RIO identifier in the **RIO Device** control. Use MAX to configure the hardware alias of the USRP RIO that you want to use as UE.
4. Select the USRP bandwidth.
5. Run "**LTE Host DL MIMO.gvi"** by clicking the run button ( ).
    - If successful, the **FPGA Ready** indicator lights.
    - If you receive an error, check if your RIO device is connected properly.

6. Set **UE RX Frequency [Hz]** to a frequency supported by your NI USRP RIO.
7. The RF configuration is in the front panel of UE/RX advanced tab. Set the RF port of UE to RX2/RF0 to act as RX1. The RX2 will be configured to RX2/RF1 automatically in case of using MIMO flag.
8. Enable the UE Receiver, which implements a Downlink Receiver, by setting the switch control to **On.**
    - If successful, the **UE RX Active** indicator lights.
9. If the control of **MIMO flag** presented in Figure 4 is enabled**,** both RX chains in the UE are active otherwise the system works as SISO.

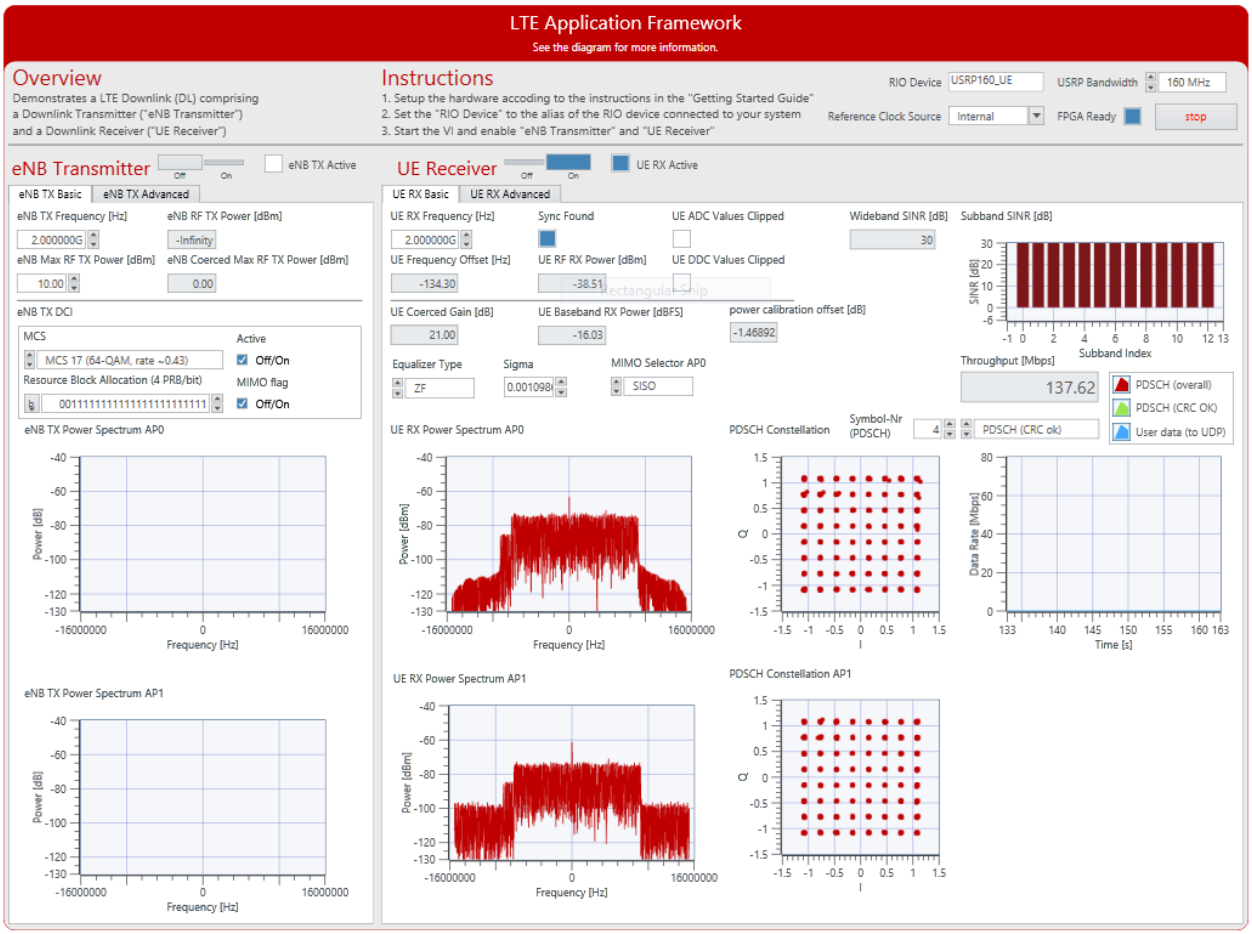

Figure 5: Front panel of UE (UE RX Basic Tab)

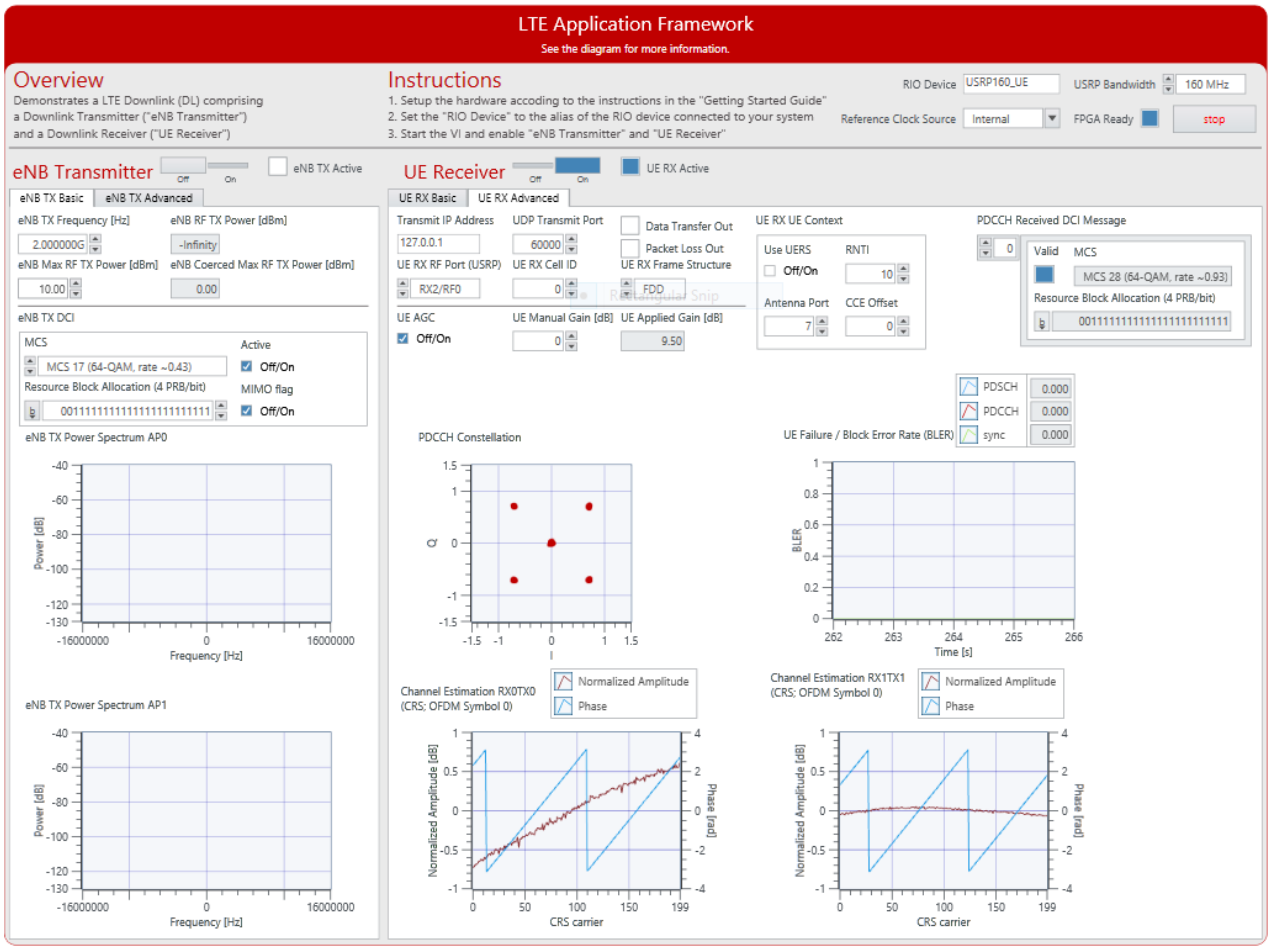

Figure 6: Front panel of UE (UE RX Advanced Tab)

Running Video Streaming

The LTE Application Example allows you to transmit video stream if you use a video streaming application as data source and a video player as data sink. For example, you can use the VLC media player, which is available at www.videolan.org.

The previous section described how to use the LTE MIMO Application Example with the double-device setup in eNodeB/UE configuration for establishing a LTE MIMO downlink link between the eNodeB device and the UE device. A basic MAC implementation allows for **packet-based data exchange** of user data between eNodeB and UE. The user data can be provided to the host using UDP as shown in Figure 7. Any program capable of transmitting UDP data can be used as a data source. Similarly, any program capable of receiving UDP data can be used as a data sink.

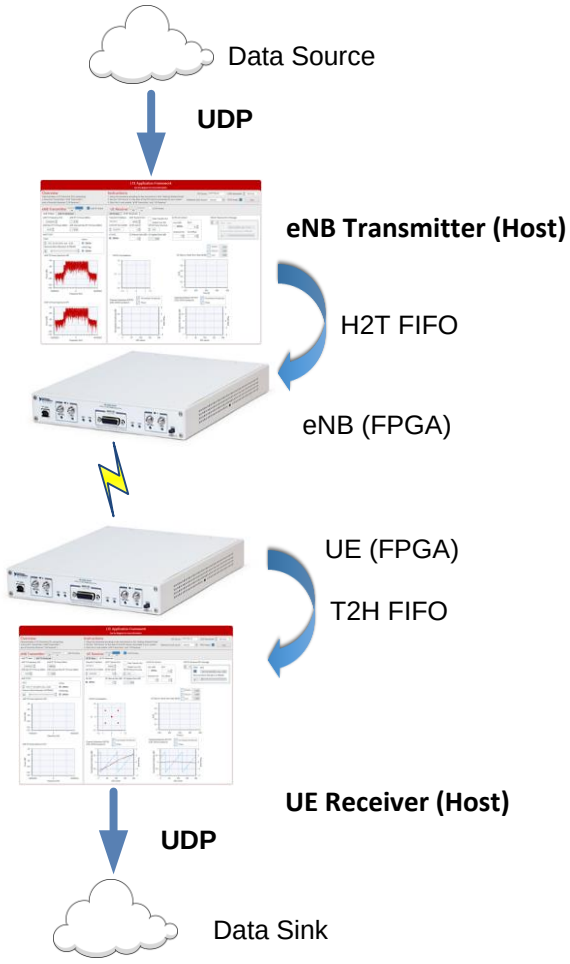

Figure 7: Data streaming from the system using UDP (eNodeB to UE)

Start Video Stream Transmitter

In the following, the commands are described that need to run on the host computer running the **LTE Host DL MIMO.gvi** (eNB Transmitter)**.**

1. Start cmd.exe and change the directory to the VLC installation directory.
2. Start the VLC application as a streaming client with the following command:
   vlc.exe --repeat "*;PATH_TO_VIDEO_FILE*"
   :sout=#std{access=udp{ttl=1},mux=ts,dst=127.0.0.1:50000} where *PATH_TO_VIDEO_FILE* should be replaced with the location of the video that should be used.

Port 50.000 is the default UDP Receive Port. For ease of use, you can also save this command line to a batch file, e.g. Stream Video LTE.bat.

Start Video Stream Receiver

In the following, the commands are described that need to run on the host computer running the **LTE Host DL MIMO.gvi** (UE Receiver)**.**

1. Start cmd.exe and change the directory to the VLC installation directory.

2. Start the VLC application as a streaming client with the following command:
   vlc.exe udp://@:60000

Port 60.000 is the default UDP Transmit Port. For ease of use, you can also save this command line to a batch file, e.g. Play Video LTE.bat.

Description of the Controls and Indicators on the Host Panel

This section provides information about the extra controls and indicators compared to the LTE Application Framework 2.0.1. Table 1 and Table 2 show the required controls and indicators of MIMO setup in the front panel, respectively.

Table 1. MIMO Controls in eNodeB and UE Front Panels

Control	Description
MIMO flag	Activates the eNodeB to run in MIMO setup. The MIMO flag is included in the DCI message to be deciphered at the RX part of UE.
Equalizer Type	Configures the algorithm that is used for equalization. The enumeration contains the following values: MMSE, ZF, and Simple.
Sigma	Selects the scaling factor that is used in the MMSE channel equalizer.
MIMO Selector AP0	Determines the PDSCH Constellation of AP0 that has been derived from SISO equalizer or MIMO equalizer.

Table 2. MIMO Indicators in UE Front Panel

Indiactor	Description
eNB TX Power Spectrum AP0	Shows the power spectrum of the DL TX baseband signal transferred to the RF0/TX1.
eNB TX Power Spectrum AP1	Shows the power spectrum of the DL TX baseband signal transferred to the RF2/TX1.
UE RX Power spectrum AP0	Shows the power spectrum of the DL RX baseband signal received from RF0/RX2. If synchronization is successful based on the received signal from RX1 (RF0/RX2), frame timing and frequency offset correction are applied on the received signals from both ports RF0/RX2 and RF1/RX2.
UE RX Power spectrum AP1	Shows the power spectrum of the DL RX baseband signal received from RF1/RX2.
PDSCH Constellation of AP0	Constellation of RX IQ samples of AP0 allocated for PDSCH transmission after SISO or MIMO equalization. Only samples for the configured OFDM **Symbol-Nr (PDSCH)** are displayed.
PDSCH Constellation of AP1	Constellation of RX IQ samples of AP1 allocated for PDSCH transmission after MIMO equalization. Only samples for the configured OFDM **Symbol-Nr (PDSCH)** are displayed.
Channel Estimation of AP0	Graphical representation of the normalized channel amplitude and phase estimated based on the cell specific reference signals of AP0
Channel Estimation of AP1	Graphical representation of the normalized channel amplitude and phase estimated on the cell specific reference signals of AP1.

System Performance

If the MCS in the eNodeB has been configured to MCS number 28 (64 QAM, rate ~ 0.93), the achieved throughput is 137.62 Mbps as it is shown in Figure 5. If the Resource Block Allocation (4PRB/bit) in the eNodeB TX Basic tab has been configured to all of ones, the achieved throughput is 150 Mbps.

Implementation Details

The main goal of LTE MIMO Application Example is to show the principle of MIMO implementation. It is implemented for DL only. The objective is to bring the LTE Application Framework to a parallel 2x2 transmission. The data throughput that has already conceived in the LTE Application Framework (75 Mbps) can be doubled by using an additional parallel antenna transmission (150Mbps). The required modifications have been made in a way that the application example should use the existing blocks with some exception in the channel Equalization and the interfaces between Host and FPGA.

Although the Host data transmission protocol is not modified. The payload is separated at the beginning and interleaved at the end in the FPGA. A number of important changes have been made in the rest of the code related to the inclusion of the MIMO flag, which can be controlled from the front panel. The MIMO flag goes through the Host parameter transmission path and ends in the FPGA TX part. This parameter leads to some modifications in the FPGA design and it is included in the DCI message to be deciphered at the RX part. In the RX part, it represents the key of how the received signals are processed. The MIMO flag basically conditions the system's FPGA processing to behave like a SISO or a MIMO system.

The TPC field in the DCI message that is reserved for uplink transmit power control commands is not used in the LTE Application Framework 2.0. Therefore, the MIMO flag is used instead of the TPC in the DCI message. For more information about the DCI message, refer to the **white paper** of LTE Application Framework.