Student Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Class\_\_\_\_\_\_\_\_

 **Bandwidth** Lesson 6

***Bandwidth*** is the range of frequencies “occupied” by a signal – in particular, the width of the range of frequencies. Recall, that when looking at a signal in the frequency domain, we saw that the signal may not be only confined in a narrow range of frequencies, but may have energy on a wider set of frequencies. See below:



Let us take a closer look at the top graph showing the “Power of the radio spectrum”



Each “point” on the frequency axis is 100 kHz. We can see that this signal occupies 400 kHz in total. This is its ***bandwidth*.**

Think of bandwidth as a garden hose we use to water our flower garden. The greater the diameter of the garden hose, the greater amount of water the garden hose can carry. Similarly, the flow of information data is affected by its bandwidth. The greater the bandwidth, the faster we can send data and the more data we can send.

***Data rate*** is how much information we can send over the air in a given amount of time. It is usually measured in *bits per second,* (a bit is a 0 or 1 value!) meaning how many bits can be transmitted within a single second.

The relationship between ***data rate*** and ***bandwidth*** is defined as follows: if all other factors are kept constant, the rate at which we can send data on an electromagnetic wave is *proportional* to the bandwidth it occupies. In other words, if we double the signal bandwidth, the rate at which this signal can carry information will also be doubled!

We are now going to perform an activity to see the relationship between data rate and bandwidth. We’ll send information using wave signals with different bandwidths, and see how long it takes to send a fixed amount of data.

**Activity**

For this activity, we are going to use a wireless testbeds that is currently available: the **WITest Testbed at NYU**. We will log in to it over the Internet and run an experiment via remote access.

Your teacher will log you into the test bed.

Then open 3 terminals, and in the first one, enter the following commands to **turn on** the nodes:

ssh witest

omf-5.4 load -i gr-nyquist.ndz -t omf.witest.node16,omf.witest.node17
omf tell -a on -t omf.witest.node16,omf.witest.node17

Still in the first terminal, log in to your **transmitter** by entering:

ssh witest

ssh root@node16

And in the second terminal, log in to your **receiver**:

ssh witest

ssh root@node17

In a third terminal, run:

ssh -L 8100:node17:8100 -L 8101:node17:8101 witest

where “witest” is login info. i.e. “geni-username@witestlab.ploy.edu”

Everybody should be up and running now.

On the **receiver** node, run:

cd ~/shinysdr
# Run shiny server
python -m shinysdr.main ~/.shiny

This last command should start a signal visualization server, which, when it is running successfully, will say something like:

INFO:shinysdr:ShinySDR is ready.
INFO:shinysdr:Visit http://localhost:8100/ShinySDR/

In a Google Chrome **Incognito** browser window. Open the URL that is shown in the Shiny server output. (<http://localhost:8100/ShinySDR/> in this example).

Configure your display as follows:

* Click on the "hamburger" icon in the top left corner to open the menu, if it isn't already open.
* Un-select the "Frequency DB" display to hide that display if it is showing.
* Select the "Radio Config" display if it isn't showing.
* In the "Radio Config" section, set the "RF Source" to "OsmoSDR", the "Antenna" to "TX/RX", and the "Center Frequency" to 2,480,000,000 (2.48 GHz).
* Note the "Gain" slider in the "Radio Config" section. You may have to increase the gain later if you aren't able to see the transmission in the ShinySDR window.
* Check the "Use DC Cancellation" box, but don't worry if it doesn't stay checked. This setting is not strictly required.
* Use the "Options" section to adjust the dynamic range and reference level of the visualization so that you can see the noise floor and there is about 60-80 dB of range (the difference between “highest value” and “lowest value” should be 60-80).

Click on the "hamburger" again to close the menu.

Next, to generate a signal, we will run the following command (all on one line!) in the transmitter terminal window:

time /usr/local/share/gnuradio/examples/digital/narrowband/benchmark\_tx.py -f 2480e6 -r 0.5e6 -M 5 -p 2 --excess-bw=0.05

Where:

* -f is used to set the carrier frequency at which to transmit,
* -r specifies the data rate at which to transmit, in bits per second,
* -M specifies the total amount of data to transmit, in megabytes,
* -p and --excess-bw relate to the modulation type (i.e. how many bits you send at a time, e.g. 1, 22, 23…)

When we run this, we are sending 5 megabytes (MB) of data and transmitting at a rate of 0.5 MB per second [This is ½ million bytes per sec.]. Look at the graph and measure how much bandwidth this uses up. Record it below

Bandwidth: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MHz

When the transmission is over look at the bottom left of the screen and see “real”. This tells the **“real time”** (in minutes & seconds) it took to transmit this data. Record the real time below

Real time: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Now let’s send another signal, but this time we are going to change the transmission rate. This time we are going to transmit at a rate of 2 MB per second [This is 2 million bytes per sec.]. In the **transmitter** terminal run the following command:

time /usr/local/share/gnuradio/examples/digital/narrowband/benchmark\_tx.py -f 2480e6 -r 2e6 -M 5 -p 2 --excess-bw=0.05

Look at the graph and measure how much bandwidth this uses up. Record it below

Bandwidth: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MHz

When the transmission is over look at the bottom left of the screen and see “real”. This tells the **“real time”** (in minutes & seconds) it took to transmit this data. Record the real time below

Real time: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Questions**

How was the bandwidth affected when we changed the transmission rate?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Compare the time it took to transmit both signals and state which was faster.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Based on everything you just did what is the relationship between the data transmission rate and bandwidth?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_