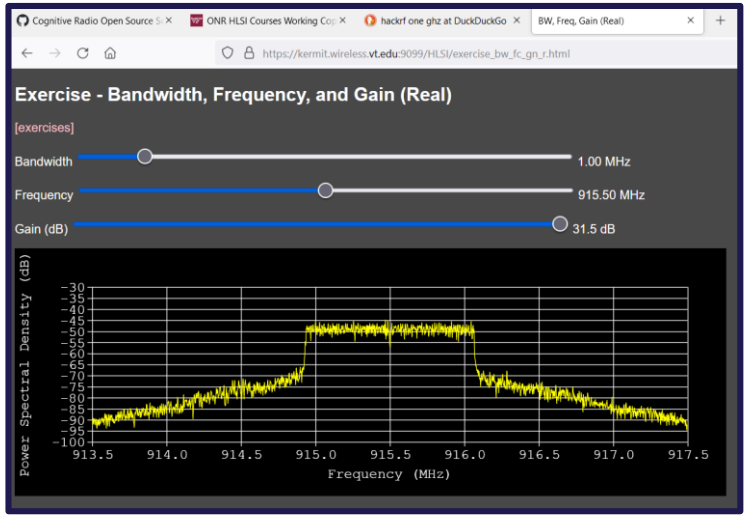
Planned Remote Lab Exercises and Simulations

11/22/2021

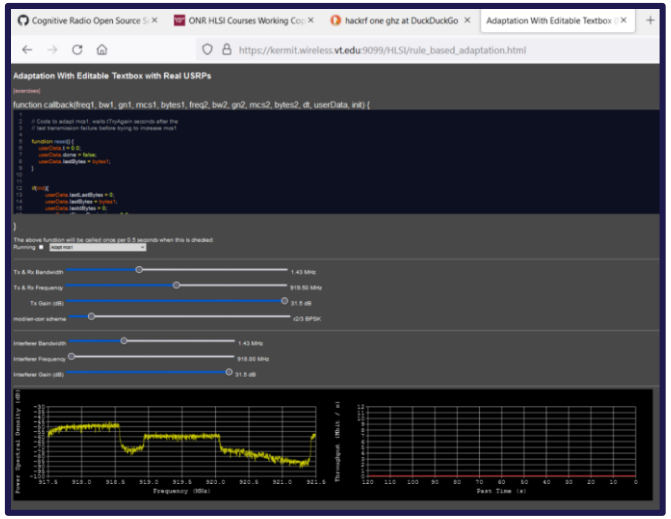
Current Remote Lab Exercises

Exercises run on Cognitive Radio Network (CORNET) testbed

Remote Laboratory Exercises using SDR Testbed (To be integrated into online learning modules)

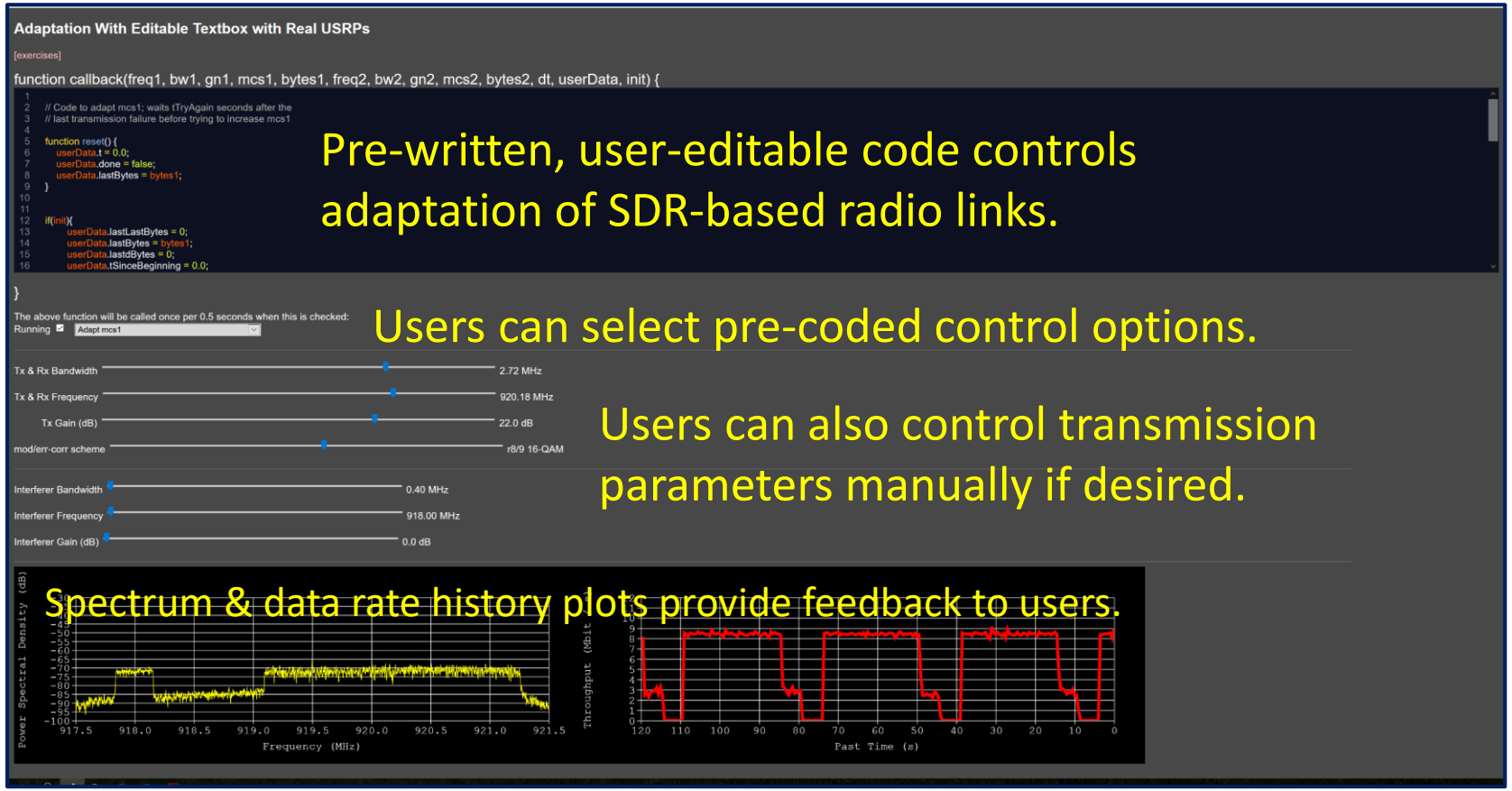


Effect of transmission parameters and nonlinear amplifier response on signal spectrum



Manual and/or editable automatic adaptation of transmission parameters to maximize data rate

Example Remote Laboratory Exercise



Remote Lab Demonstration

Future CRTS* Competitive/ Collaborative Spectrum-Sharing Remote Lab Exercises

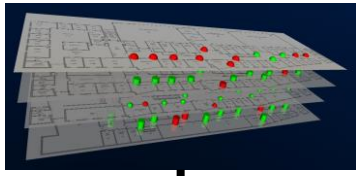
*Cognitive Radio Test System, a software framework for managing experiments on a wireless testbed (development in progress)

Multi-Team CRTS Spectrum-Sharing Exercise

- User interface will work similarly to the UI for the current CRTS exercise *exercise_interference_throughput_mode_r.html*
- However, each team of one or more students will be given access to a separate UI
- Each team's UI will include
  - Sliders that only control that team's Tx & Rx
  - An editable callback that only has access to parameters of that team's Tx and/or Rx
- Each team's UI should display:
  - Throughput history for all team's radio links, plotted on the same graph using different colors or line types
  - Total number of bits successfully transmitted and received by each team during exercise
- Support for more than two teams would be ideal, but not if it takes much more time to implement

Required radios:

- *One transmitter and one receiver per team*
- *One receiver to monitor the spectrum*
- ***All can be same type** (USRP, etc.)*

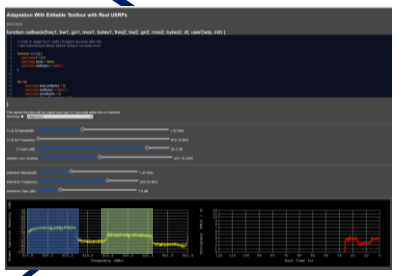


SDR-based wireless testbed

Internet

Team 1's Adaptation Code

Team 2's Adaptation Code



Spectrum

Data Rate History

Team 1

Team 1's Signal

Team 2's Signal

Team 2

...

...

Notional GUI for Two-Team Spectrum Sharing Exercise

Spectrum Sharing (Two Teams): ***Team 1***

[exercises]

```
function callback(freq1, bw1, gn1, mcs1, bytes1, freq2, bw2, gn2, mcs2, bytes2, dt, userData, init) {
  1   freq2 += 0.04e6;
  2   if(freq2 > max_freq2)
  3     freq2 = min_freq2;
  4
  5   return { freq2: freq2 };
}
```

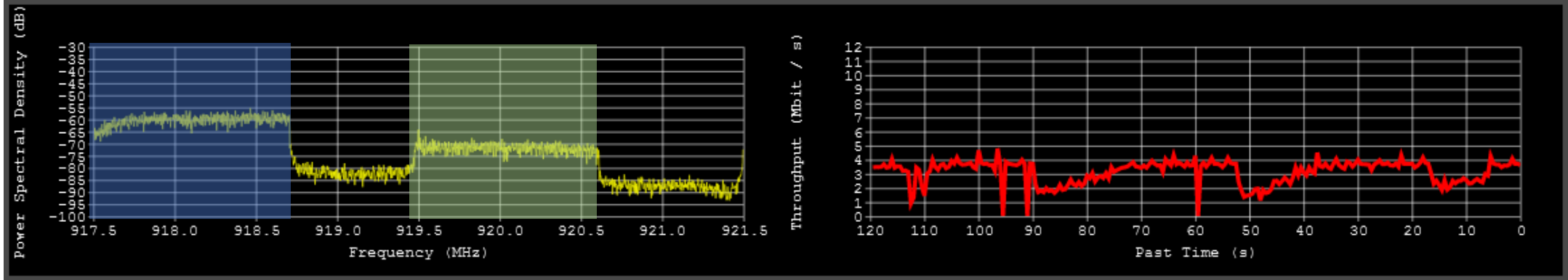
The above function will be called once per 0.5 seconds when this is checked:
Running ☑ Changing freq2

Tx & Rx Bandwidth — 1.52 MHz

Tx & Rx Frequency — 918.10 MHz

Tx Gain (dB) — 26.5 dB

mod/err-corr scheme — r2/3 16-QAM



Team 1 Signal

Team 2 Signal

Team 1 Total Bits Sent and Received: 251.2 Mbits

Team 2 Total Bits Sent and Received: 598.3 Mbits

Team 1 Score: **1,100.7**

Team 2 Score: **1,447.8**

Future CRTS RF Front End (RFFE)-Aware Spectrum-Sharing Remote Laboratory Exercises

RF Front End (RFFE)-Aware Spectrum-Sharing Exercises

- User interface will work similarly to the UI for the current CRTS exercise *exercise_interference_throughput_mode_r.html*
- Student will control frequency assignments and power for multiple signals, using sliders and a callback function
- Bandwidth can be narrow and fixed
- UI control:
  - Sliders to set frequency, power, and/or MCS for each comm link (Tx/Rx pair)
  - An editable callback that can
    - Get and set frequency, power, and/or MCS for each comm. link (Tx/Rx pair)
    - Get current data rate for each comm. link
- UI display:
  - Throughput history for all comm links, plotted on the same graph using different colors or line types
  - Total number of bits successfully transmitted and received by each link and by all links during exercise
- Multiple variations to introduce concept to students:
  - Two comm links using receivers (Rx) that have higher-performance RF front ends (RFFEs)
  - Two comm links using Rx that have lower-performance RFFEs
  - 3-4 comm links that include some higher-performance and some lower-performance RFFEs

Required radios:

- *One transmitter and one receiver per comm link*
- *One receiver to monitor the spectrum*
- ***Mix of SDR models** and/or Rx settings to yield different RFFE performance*

Notional GUI for RFFE-Aware Spectrum-Sharing Exercise

Alternatives to USRP N210 include Adalm Pluto, RTL-SDR, USRP N210 with different Rx Gain Settings (if this provides sufficiently different performance)

Spectrum Sharing (RFFE-Aware, Two Comm. Links, USRP N210s)

[exercises]

```
function callback(freq1, bw1, gn1, mcs1, bytes1, freq2, bw2, gn2, mcs2, bytes2, dt, userData, init) {
    freq2 += 0.04e6;
    if(freq2 > max_freq2)
        freq2 = min_freq2;

    return { freq2: freq2 };



}
```

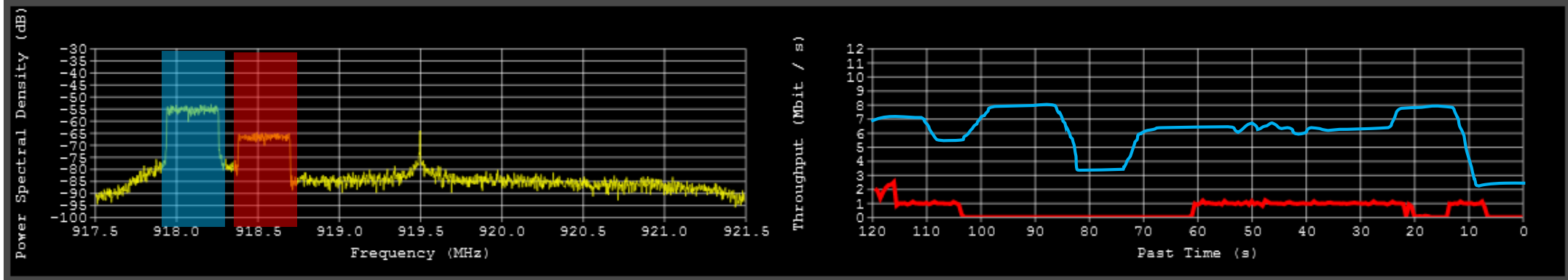
The above function will be called once per 0.5 seconds when this is checked:
Running ☑ Optimize spectral efficiency

Comm Link 1

Tx & Rx Bandwidth	1.52 MHz
Tx & Rx Frequency	918.10 MHz
Tx Gain (dB)	26.5 dB
mod/err-corr scheme	r2/3 16-QAM

Comm Link 2

Tx & Rx Bandwidth	1.52 MHz
Tx & Rx Frequency	918.10 MHz
Tx Gain (dB)	26.5 dB
mod/err-corr scheme	r2/3 16-QAM



Comm Link 1 (USRP N210 Rx)
Comm Link 2 (USRP N210 Rx)

Link 1 Spectral Efficiency: **0.59 Mb/s/Hz**
Link 2 Spectral Efficiency: **0.02 Mb/s/Hz**

Overall Spectral Efficiency: **0.31 Mb/s/Hz**

Notional GUI for RFFE-Aware Spectrum-Sharing Exercise using Adalm Pluto SDRs

Throughput expected to be lower than if using USRP N210s for close frequency spacing

Spectrum Sharing (RFFE-Aware, Two Comm Links, Adalm Pluto SDRs)

[exercises]

function callback(freq1, bw1, gn1, mcs1, bytes1, freq2, bw2, gn2, mcs2, bytes2, dt, userData, init) {

```
1   freq2 += 0.04e6;
2   if(freq2 > max_freq2)
3      freq2 = min_freq2;
4
5   return { freq2: freq2 };
```

}

The above function will be called once per 0.5 seconds when this is checked:

Running ☑ Use minimum freq. spacing

Comm Link 1

Tx & Rx Bandwidth 1.52 MHz

Tx & Rx Frequency 918.10 MHz

Tx Gain (dB) 26.5 dB

mod/err-corr scheme r2/3 16-QAM

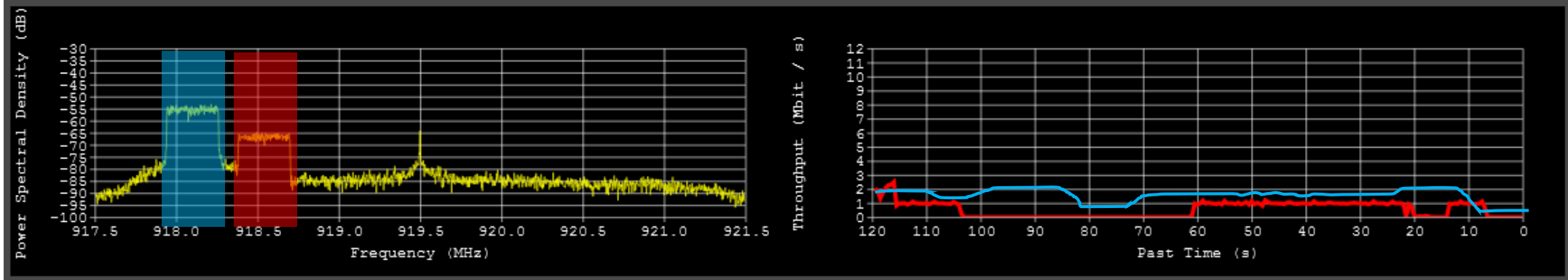
Comm Link 2

Tx & Rx Bandwidth 1.52 MHz

Tx & Rx Frequency 918.10 MHz

Tx Gain (dB) 26.5 dB

mod/err-corr scheme r2/3 16-QAM



Comm Link 1 (Adalm Pluto SDR Rx)

Comm Link 2 (Adalm Pluto SDR Rx)

Link 1 SE: **0.05 Mb/s/Hz**

Link 2 SE: **0.02 Mb/s/Hz**

Overall Spectral Efficiency: **0.03 Mb/s/Hz**

Notional GUI for RFFE-Aware Spectrum-Sharing Exercise using more than one type of SDR as receiver

Spectrum Sharing (RFFE-Aware, Two Comm Links, Adalm Pluto and USRP N210 SDRs)

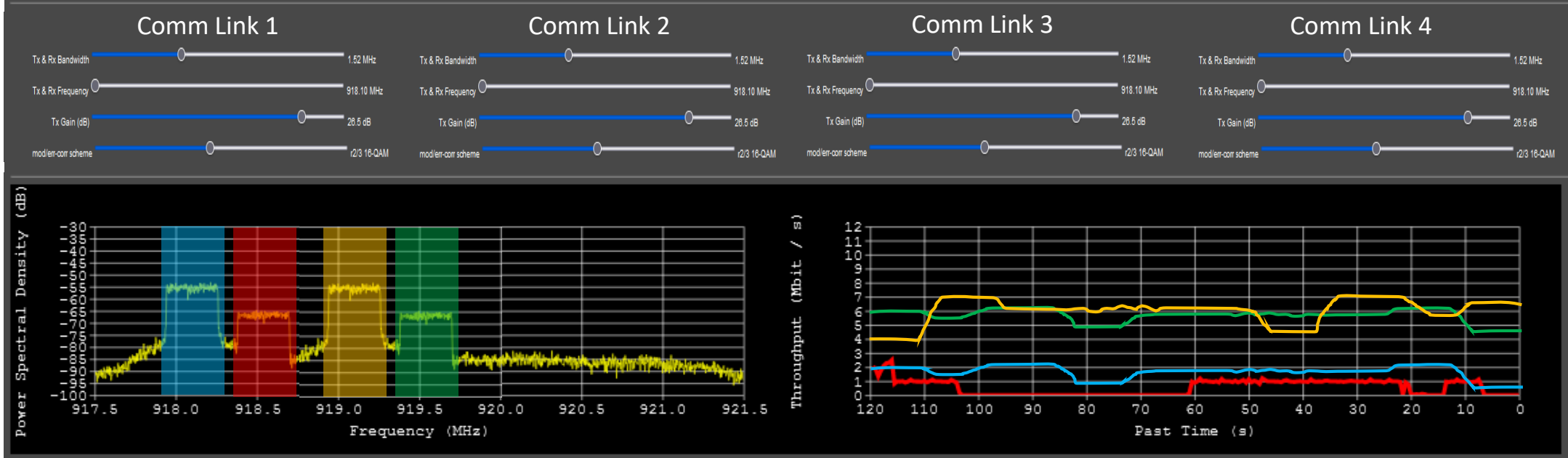
[exercises]

```
function callback(freq1, bw1, gn1, mcs1, bytes1, freq2, bw2, gn2, mcs2, bytes2, dt, userData, init) {
    freq2 += 0.04e6;
    if(freq2 > max_freq2)
        freq2 = min_freq2;

    return { freq2: freq2 };

}
```

The above function will be called once per 0.5 seconds when this is checked:
Running ☑ Optimize spectral efficiency



Comm Link 1 (Adalm Pluto SDR Rx)
Comm Link 2 (Adalm Pluto SDR Rx)
Comm Link 3 (USRP N210 Rx)
Comm Link 4 (USRP N210 Rx)

SE (Mb/s/Hz): L1 **0.59**, L2 **0.02**
L3 **0.06**, L4 **0.02**, Total **0.17**

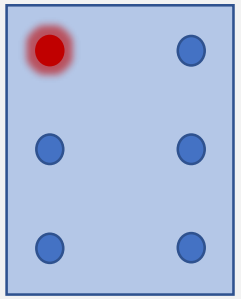
Notional CRTS Administrator Menus

Scenario
Single Signal
Comm Link + Noise
Comm Link + Interferer
Spectrum Sharing (w/ Auto Comm Link)
Spectrum Sharing (Two Teams)
…

Scenario Parameters

f_min (MHz)	910.000
f_max (MHz)	912.500
User 1	team1
User 2	team2
GUI 1 URL	https://HLSI.vt.edu/t1
GUI 2 URL	https://HLSI.vt.edu/t2

Table and/or Map of SDRs
SDR 1
SDR 2
…
SDR N



Role in Scenario for SDR 1 (10.10.0.1)

Team 1 Tx
Team 1 Rx
Team 2 Tx
Team 2 Rx
Spectrum monitor Rx

Future HLSI* Simulations

*Hands-on Learning for Radio Frequency Spectrum Innovation

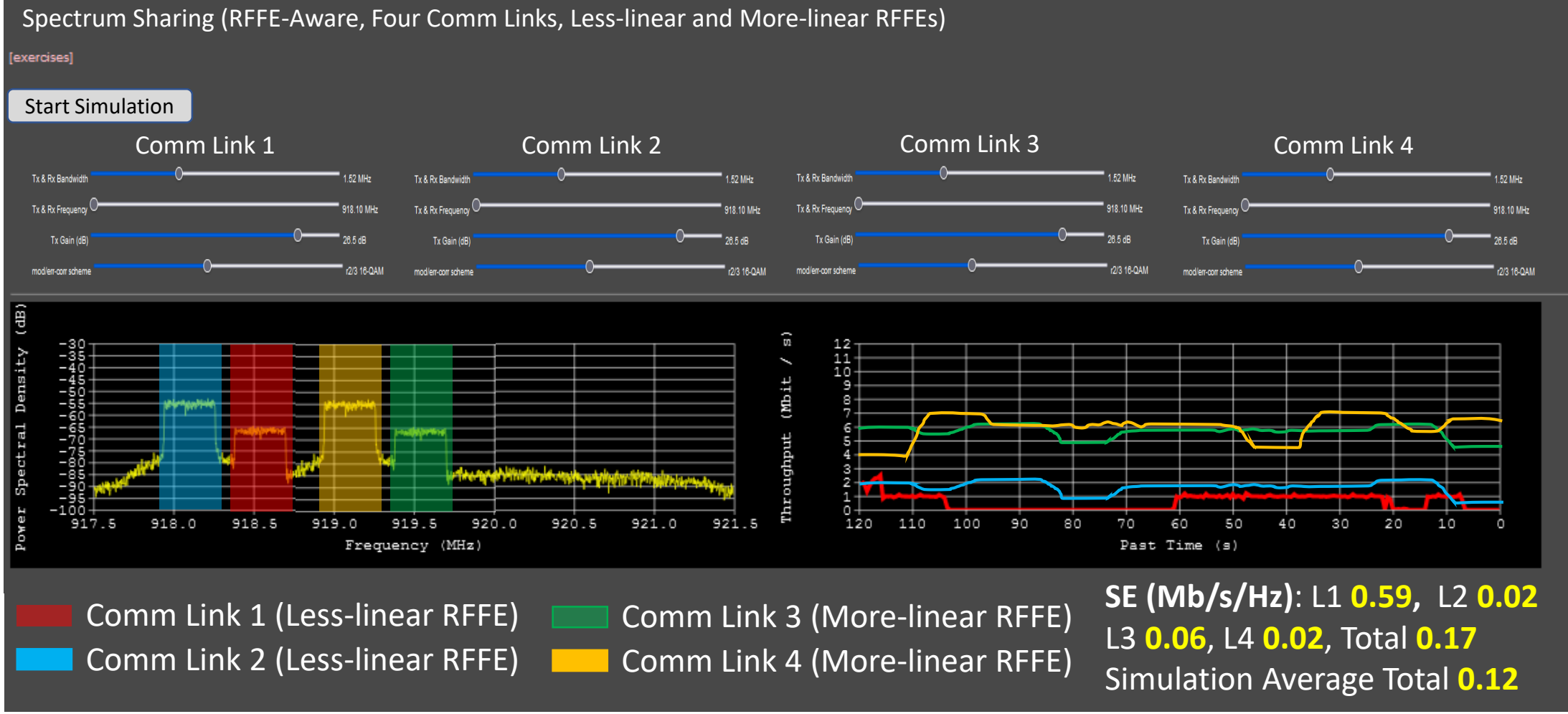
RF Front End (RFFE)-Aware Spectrum-Sharing Simulations

- User interface will work similarly to the UI for the current HLSI exercise: http://vtwireless.github.io/HLSI/13_manual_avoidance.html, except that
  - There will be no frequency-hopping interferer and no Hop Rate control
  - Frequency, Bandwidth, Gain, and Mod Code controls will be provided for multiple signals (two to four or more signals, depending on the version of the exercise
  - Comm link metrics such as Capacity or Actual (simulated) Data Rate, Spectral Efficiency, and Total Bits Transmitted & Received during the Simulation will be provided for each signal
  - Cumulative statistics for Capacity, Spectral Efficiency, and Actual Data Rate will be displayed.
- The simulation will be timed so that it can be used in a competitive manner
- The initial version of the exercise will use the same calculations for comm link metrics as those used in https://vtwireless.github.io/HLSI/13_manual_avoidance.html.
- Future versions of the exercise will calculate comm link metrics based on work by Prof. Jeff Reed and his research team on spectrum sharing that takes into account receiver RF front-end performance

Simulated radios:

- ***One transmitter, one receiver per comm link***
- ***Mix of receiver models*** *that have different modeled RFFE performance*
- ***Modeling may be high-level rather than detailed****: Some hardware characteristics may be abstracted or implied*

Notional GUI for RF Front End (RFFE)-Aware Spectrum-Sharing Simulations with Manual Frequency Assignment*



UI Display

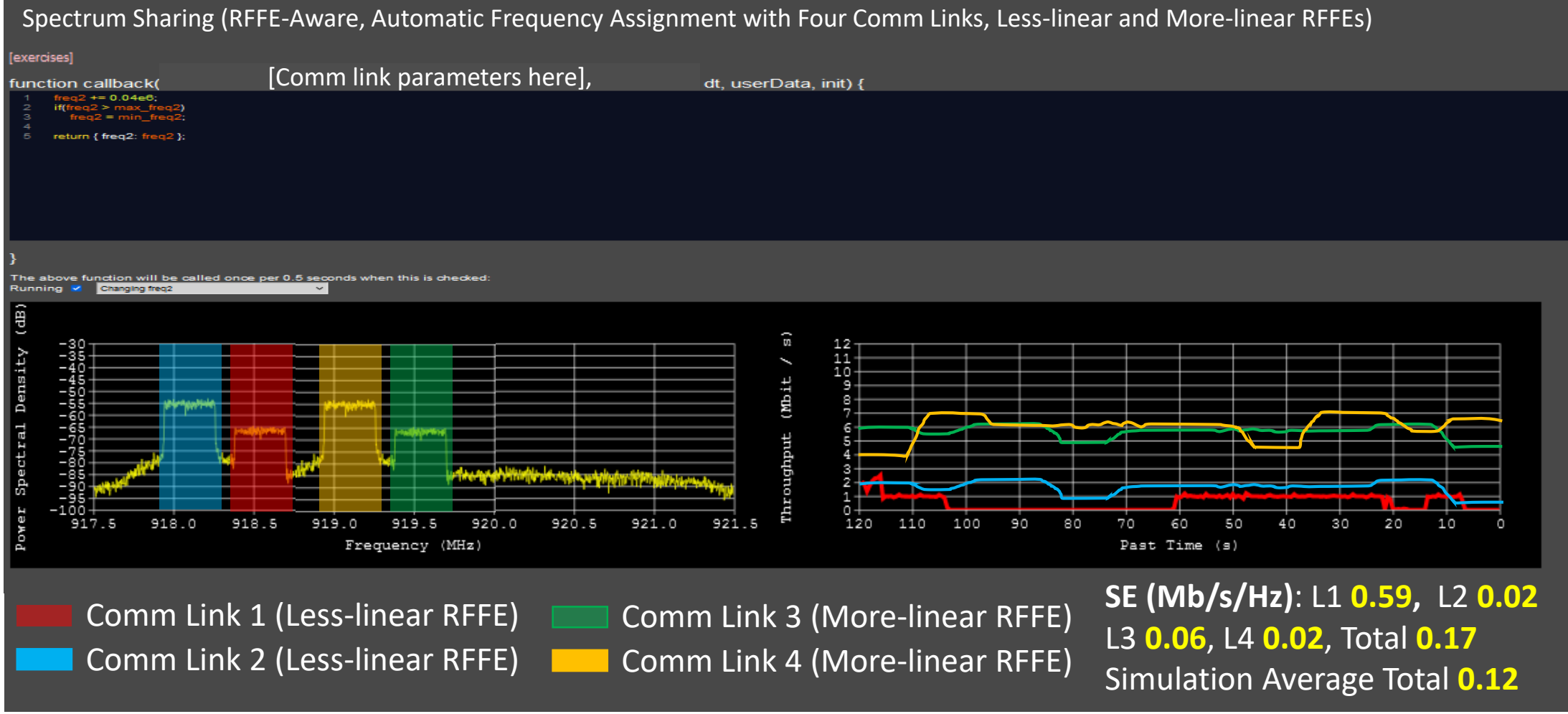
- Throughput history for all comm links, plotted on the same graph using different colors or line types
- Total number of bits successfully transmitted and received by each link and by all links during exercise

Multiple variations to introduce concept to students

- Two comm links using receivers (Rx) that have higher-performance RF front ends (RFFEs)
- Two comm links using Rx that have lower-performance RFFEs
- Four or more comm links that include some higher-performance and some lower-performance RFFEs

***Note**: Lines in spectrum plot will appear smoother and data throughput changes will appear more abrupt in simulations.

Notional GUI for RF Front End (RFFE)-Aware Spectrum-Sharing Simulations with Automatic Frequency Assignment



Similar to manual frequency assignment simulation, except:

- Editable callback function will replace or augment slider controls
- Callback will be able to read link and overall metrics and set link parameters

Multiple variations to introduce concept to students

- Two comm links using receivers (Rx) that have higher-performance RF front ends (RFFEs)
- Two comm links using Rx that have lower-performance RFFEs
- Four or more comm links that include some higher-performance and some lower-performance RFFEs