

ECE 6110 Lab Assignment 4: Wireless Propagation and Rate Adaptation

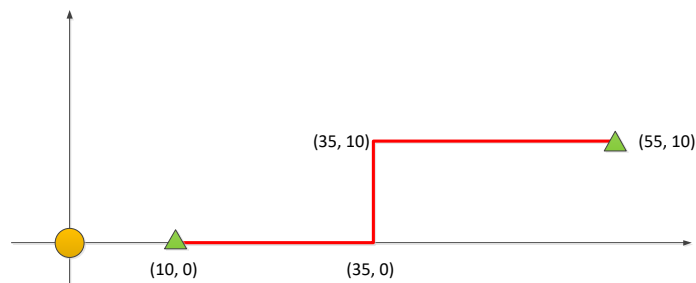
In this lab, you will investigate how the Minstrel rate adaptation algorithm adapts rates as user location changes, thereby affecting the received signal strength, and you will also analyze the impact of different wireless propagation models.

Turn-in Instructions: Create a separate folder for each part of the assignment inside a main folder named “Lab4_<your_last_name>_<your_first_name>”. Then create a single tarball of the main folder and all of its subfolders and submit it in Canvas.

Important programming instructions: All code must be hand typed in your editor of choice. No auto-formatting is allowed to change the text that you type in manually. Also, make sure to use the exact program name and command-line parameter names specified in each part. You will get points deducted if you do not follow these instructions even if your code is functionally correct!!

Part 1: Trace Rate Changes and Goodput as User Location varies

In this part, you will build on Lab 3, Part 2 to trace in more detail what happens as user location varies with respect to a fixed access point. As in Lab 3, you should use the 802.11ac standard, the Minstrel Ht WiFi manager, a PacketSink application on the wireless station (user device), and an OnOff application on the access point. In this lab, however, the access point location and station movement should be set as shown below, where the yellow circle represents the access point and the green triangle represents the station:



One very important difference from Lab 3 is that, in order to trace rate changes properly, you should collect data at all locations in a **single simulation run**. You can use the following program as an example to see how to accomplish this:

```
$NS3/examples/wireless/wifi-rate-adaptation-distance.cc
```

This program also illustrates how to track the rate changes Minstrel makes as signal strength varies during the simulation. Parameters of the simulation should be set as follows:

- OnOff application: rate of 400 Mbps and packet size of 1024 bytes
- simulation time at each location: 1 second
- step size of station location movement: 5 meters
- height of access point and height of station: 3.0 meters and 1.0 meters, respectively (this will allow the 2-ray ground model to be used in Part 2 of the lab)

Your program should output the distance from the access point and goodput at each station location and all rate changes that occur during the simulation run. These outputs should be combined in a single output stream with the simulation times for each line of output so you can see how the rate changes impact the resulting goodput. There are no command-line parameters for the program of this part. Your code will just execute a single simulation run with parameters as defined above.

Turn-in instructions: In your Part 1 subfolder, include:

- 1) the source code of your program (name your program “lab4-part1.cc”);
- 2) a screen shot showing the command line and output for your program;
- 3) a text file containing the goodputs and rate changes output of your program;
- 4) a few sentences discussing how the rate changes exhibited by Minstrel impact the goodputs measured at different locations.

Part 2: Evaluation of Different Path Loss Models

In this part, you will evaluate and compare different wireless propagation models, specifically the two-ray ground model and the Nakagami fading model. You will use the same setting and parameters as in Part 1 but simply change the propagation model in the following way:

```
YansWifiChannelHelper wifiChannel;  
wifiChannel.SetPropagationDelay ("ns3::ConstantSpeedPropagationDelayModel");  
wifiChannel.AddPropagationLoss ("ns3::XXXLossModel");  
wifiPhy.SetChannel (wifiChannel.Create ());
```

where “XXX” is replaced by either “Nakagami” or “TwoRayGround”. Choose the propagation model using a command-line parameter named “propagation”. You should also add the following parameters and corresponding values to the AddPropagationLoss() method invocation for the Nakagami model:

- Distance1: 35.0
- Distance2: 40.0
- m1: 0.25
- m2: 0.11

Output the goodput and distance at each location and Minstrel rate changes, as in Part 1, but also generate a line of output each time a packet is dropped by the application. You can do this by using the “PhyRxDrop” trace source with a config path as detailed in the ns-3 Doxygen documentation on the WifiPhy class:

https://www.nsnam.org/docs/release/3.35/doxygen/classns3_1_1_wifi_phy.html

and a callback function prototype as follows:

```
static void RxDrop (std::string context, Ptr<const Packet> p, WifiPhyRxfailureReason reason)
```

Turn-in instructions: In your Part 2 subfolder, include:

- 1) the source code of your program (name your program “lab4-part2.cc”);
- 2) goodput vs. distance plots for both Nakagami and two-ray ground models with one data point for each wireless station location;
- 3) a screen shot showing the command line and output for your program with the two-ray ground model;
- 4) a text file containing the output of the program for the Nakagami model; and
- 5) a few sentences comparing and contrasting the packet loss behavior of the two models and explaining what you think the reason for the different behaviors is.