

# LAB6: Guidance to Final Projects

## CS169: Mobile Wireless Networks - Winter 2017

Kittipat Apicharttrisorn (Patrick)

Department of Computer Science and Engineering  
University of California, Riverside

February 27-28, 2017

# Table of Contents

- 1 How to setup network topology?
- 2 How to saturate WiFi traffic?
- 3 How to count number of packet loss or packet loss rate?
- 4 How to measure throughput?
- 5 How to measure delay caused by backing off?
- 6 How to change the exponential backoff to linear backoff?
- 7 Proposals how to mix MS and hpMS in the same simulation

# How to setup network topology?

- copy examples/wireless/simple-ht-hidden-stations.cc to scratch folder
- Try to understand what it is doing and to run it with different configurations
- Increase number of station nodes (MS) to 4
- Create MS and AP locations according to the project paper
- Install four UDP echo server applications on the AP node
- Install four UDP echo client applications on each MS node connecting to each corresponding server application

## Warnings

UDP *echo* client and server, not just UDP client and server

# How to saturate WiFi traffic?

- Set attributes - MaxPackets and Interval
- Interval = 0.1 means 10 packets per second
- Keep MaxPackets to infinity and simulation will terminate when time is over
- Create a command line argument (string) to make it more convenient to change interval
- Decrease packet interval until packet loss rate is approximately 20% (see next slides)

# How to count number of packet loss or packet loss rate?

- Create two trace sink functions to get notified when clients have sent and servers have received.
- Copy echo server files, created by TA, from course website to replace ones in ns-3
- They provide trace sources for udp echo servers, which is not available ns-3
- Use similar call back signature and config path to udp echo client's tracesource
- Count the number of the two events to calculate packet loss rate
- Use *context*'s string to distinguish between each client's traffic

# How to measure throughput?

- Count the number of packets received by each server application (*numPacketServerReceived*)
- throughput (Mbps) =  $\frac{(\text{numPacketServerReceived} \times \text{payloadsize} \times 8)}{\text{simulationTime} \times 1000000}$
- As packet interval decreases, measure the achieved throughput

# How to measure delay caused by backing off?

- Enable NS\_LOG for DcfManager component and run the script
- Take a look at dcf-manager.cc and find the line that prints out expected time for backing off
- You can use that estimation for calculating backing off delay
- You may want to format the output, save it into a file, and use excel to calculate
- Can you think of a way to distinguish outputs from different nodes?  
(trivial for us now for tracesources)

## Bonus!

You can do something in this file to count the number of collisions that cause nodes to back off.

# How to change the exponential backoff to linear backoff?

- An answer lies in DCF manager.
- You may look at a line that takes one value of contention window (as seen in Pseudocode part of the project's paper) and double it every time a collision occurs
- Change that line to a linear increase (hpMS)
- Measure everything again and compare the two backing off policies



# Proposals how to mix MS and hpMS in the same simulation

- 1 Add a new module named *dcf-manager-linear.cc* and *dcf-manager-linear.h* and force hpMSes to wire to this module
- 2 Add a new type to WiFi mac called *StaWifiMacLinear* and set it to mac that is installed to hpMSes.  
*mac.SetType ("ns3::StaWifiMacLinear", ...*
- 3 Add an attribute to DCF manager, an interger *BackoffMode* and set it to zero for exponential backoff (default) or one for linear backoff, something similar to this  
*Config::SetDefault ("ns3::RangePropagationLossModel::MaxRange", DoubleValue (5));*