This assignment will familiarise you with various aspects of the MAC layers and above in wireless communications.
As in assignment 2, you should always remember to reset your wireless drivers at the beginning of each experiment (e.g. down, then hocStart). Run all experiments in 802.11b mode.
No new files or templates are provided for this assignment: you are expected to modify files from previous assignments or create new ones yourself.
Similarly to assignment 2, each of your experiment scripts/code should generate your entire set of raw data in a single run.

1 Part A,B - TCP over wireless (8 points)

TCP is the primary transport protocol in the Internet. As such, TCP is designed to provide reliable data transfer between hosts connected by a series of wired routers. In this experiment you will study how well TCP performs over wireless links.

1.1 Experiment Description

Use 5.5Mbps fixed rate selection for both parts.
Part A: For UDP then TCP, measure the data rate between two nodes for link loss settings between -90Db and -110Db (inclusive) in 2 Db increments. Plot these data rates on a single graph.
Part B: Now measure the performance of TCP and UDP over an unstable wireless link (fluctuating periodically between 150Db loss for approximately 0.2s and 0Db loss for approximately 1.8s).

1.2 Discussion Questions

As usual, these are not essay questions. One or two sentences per question will suffice.

1. Explain briefly why TCP woud be expected to perform more poorly than UDP over wireless connections.
2. Discuss the pros and cons of using TCP in a wireless environment.
3. Name one functionality at the MAC layer that reduces the impact of wireless on TCP.

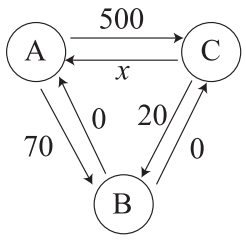

Figure 1: Link losses in Part C

1.3 Deliverables

Text file `experiment3a.txt` containing the collated data of the experiment (i.e. table containing the data rate of TCP and UDP at each loss setting). Also, text file `experiment3b.txt` containing the results of TCP and UDP on the unstable link.
Graph `experiment3a.eps` showing the data from part A in two plots (data rate of TCP, data rate of UDP) on the same graph, with the loss setting on the X-axis.
The script and java files `experiment3a.xml, experiment3a.java` and `experiment3b.xml, experiment3b.java` which generated these results.
A file `experiment3ab-discuss.txt` containing answers to discussion questions.

1.4 Hints

1. Check the iperf documentation to see how to send TCP traffic. Remember not to use the `-b` option for TCP flows since TCP determines its own data rate.

2. You may use the Thread.sleep(milliseconds) functionality to implement the unstable link (timings need not be super-accurate).

2 Part C - RTS/CTS (6 points)

The RTS/CTS mechanism is implemented in the 802.11 standards to deal with the hidden terminal problem. In part C, you will study the impact of using RTS/CTS on network performance.

2.1 Experiment Description

Deploy three nodes, A,B,C. Nodes A and C will be sending traffic to the receiving node B, with node C acting as an interference source. Node C sends UDP traffic as fast as it can to node B. Assign link losses per the diagram in Figure 1. These loss settings ensure that if node A and C transmit a packet simultaneously, the packet from A will not be received by B. Furthermore, since node C cannot hear node A, it will continuously send at its own maximum rate. We will vary A's ability to hear C's transmissions (i.e. the "hidden terminal-ness" of C) and observe the effect of RTS/CTS on A's ability to send traffic to B. Specifically, vary the

loss from C to A between 80Db and 120Db in 5Db increments, and measure the data rate from A to B. Repeat the experiment with and without RTS/CTS.

2.2 Hints

1. Use `sudo iwconfig ath0 rtsthr off` to turn off RTS/CTS and `sudo iwconfig ath0 rtsthr 10` to turn it on.

2. Remember to turn on/off RTS/CTS capability on *all three* nodes A,B,C.

2.3 Discussion Questions

1. Comment on the data rate of A with and without RTS/CTS. When is A more sensitive to its ability to hear the hidden terminal C? Why?

2. Consider A's bitrate in the "half-hidden" terminal scenario when A could hear C clearly but C could not hear A. Since A can hear C, it should already be able to avoid most collisions easily using carrier sense. Why then does RTS/CTS improve A's data rate in this situation? What would be a better solution for this functionality?

2.4 Deliverables

Data file `experiment3c.txt` containing the experiment results, i.e. the data rate of A to B for various loss settings for the link from C to A.
Graph `experiment3c.eps` containing the plot of the above.
The script and java files `experiment3c.xml, experiment3c.java` for the experiment.
A file `experiment3c-discuss.txt` containing answers to discussion questions.

3 Part D - Capture Effects (6 points)

While MAC protocols are designed to avoid collisions, a collision does not necessarily mean that both packets are lost. Depending on the wireless card, there is a chance that the packet with the stronger signal can still be received. This phenomenon is known as *c*apture. In this experiment we will investigate the effect of capture on competing senders.

3.1 Experiment Description

Similarly to part C, deploy three nodes A,B,C such that A and C send UDP traffic as fast as they can to node B. Implement a complete hidden terminal situation where A and C cannot communicate directly with each other at all and turn off RTS/CTS. Set the remaining link losses as per Figure 2.
Now vary the loss of the link from A to B between 40 and 110Db in 5Db increments. Observe how the data rates of both A and C vary with this.

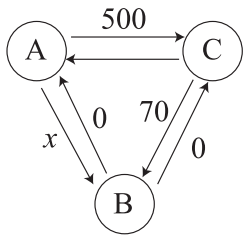

Figure 2: Link losses in Part D

3.2 Discussion Questions

1. What is the approximate difference in power between signals such that the stronger signal experiences little to no packet loss?

2. Comment on the implications of capture on spatial re-use and the network topology of dense networks with high data rates. How would this phenomenon change power level settings in dense networks? What about the MAC Clear Channel Assessment thresholds?

3.3 Deliverables

Data file `experiment3d.txt` containing the experiment results, i.e. the data rates of both A and C for various loss settings for the link from A to B.
Graph `experiment3d.eps` showing the plots derived from the data file.
The script and java files `experiment3d.xml, experiment3d.java` for this experiment.
Answers to discussion questions `experiment3d-discuss.txt`.