Final Project Description
Wireless Networking: Fundamentals and Applications

The main purpose of the final project of this course is for you to apply the concepts learned from the lectures and the tools learned from the lab assignments to solve real-world problems. You will have the freedom of selecting from a number of different topics for your final project. Most topics are related to the tools with which you are already familiar, such as software defined radio (USRP), network simulator (NS-2), and Arduino/Zigduino boards. For most topics, you can choose to implement the same system as described in the related paper. However, we encourage you to also consider additional improvements over the original design. Teams with such considerations would receive higher grades.

We understand that you might not understand the proposed topics completely from the short description in this document. If you have any question, please feel free to send us e-mails, or schedule an appointment with Michael to discuss what you'd like to do for your final project.

- **Items to be submitted, grade contribution, and deadlines for each team:**

1. **Project proposal**: (10%)
   A one-page summary describing the objectives of your project. The purpose of this summary is for us to evaluate the scope and the topic is appropriate for the final project. Please send the summary in standard ACM conference format, as a PDF file, by the due date to wn@csie.ntu.edu.tw. You will receive our comments about your proposal after we read your proposals.
   <u>Due date: 5/26 23:59</u>
2. **Final project presentation:** (40%)
   Each team will give a 15-minute presentation. The presentation should report the problem you are trying to solve, the experimental setup, and the results. You are encouraged to prepare a short video that can demonstrate your working system and play the video during your presentation.
   <u>Presentation dates: 6/16 (Mon) and 6/18 (Wed), during the usual lecture time in the final exam week</u>
3. **Final report:** (50%)

The final report should be in standard ACM conference paper format. The report should at least have the following: abstract, introduction, related works, experimental setup and results, conclusion, and references listed at the end of the paper. The report should have at least 4 pages and can be up to 8 pages. Please send the report as a PDF file to wn@csie.ntu.edu.tw by the due date.

Due date: 6/21 23:59

The template for the report can be downloaded from here:
http://www.acm.org/sigs/publications/proceedings-templates (option 2)

- **Project topics:**

1. ZigZag: https://homes.cs.washington.edu/~gshyam/Papers/ZigZag.pdf

ZigZag is a decoding scheme that allows a node to decode successive collided packets. Instead of avoid hidden terminals, it decodes packets from hidden terminals. In this project, we will implement ZigZag decoding using USRP, and evaluate the decoding success probability and SNR loss due to ZigZag decoding.

2. ZFBF: http://www.thlab.net/~thsalon/papers/aryafar10Design.pdf

Zero-forcing beamforming allows a multi-antenna AP to send packets to multiple single-antenna clients simultaneously. In this project, we will implement ZFBF in a 3-antenna AP scenario using USRP. We will evaluate using USRP how this ZFBF improves the aggregate throughput, as compared to single-user transmission w/ and w/o using transmit diversity.

3. Intra-flow network coding:
   http://groups.csail.mit.edu/netmit/wordpress/wp-content/themes/netmit/papers/MORE.pdf

Opportunistic routing is a recent technique that achieves high throughput in the face of lossy wireless links. However, multiple nodes may hear a packet broadcast and unnecessarily forward the same packet. A strict scheduling on routers' access is required to avoid redundant forwarding. This problem can be solved by intra-flow network coding, which ensures that routers that hear the same transmission do not forward the same packet. In this project, we will implement intra-flow network coding in NS2, and evaluate how it outperforms traditional ad-hoc routing in both static and dynamic environments.

4. Characterizing WiFi Traffic

http://djw.cs.washington.edu/papers/trace-ewind.pdf

http://www.cs.umd.edu/~dml/papers/fidelity_pam08.pdf

http://research.microsoft.com/en-us/um/people/srikanth/data/imc09_dcTraffic.pdf

While software defined network (SDN) becomes an emerging technology, people are thinking about realizing wireless SDN. To realize this goal, it is important to analyze the characteristics of wireless traffic for efficient traffic engineering. Though similar measurements have been done in around 2005, many new applications, such as facebook and youtube, are published after then. These new applications might significantly change the characteristics of wireless traffic. In this project, we will monitor the wireless traffic in the campus networks, and try to figure out how SDN can help improve wireless resource utilization.

5. Camera Communications

UFSOOK: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6676454

RollingShutter Sampling:

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6477759

Camera Communications is a form of visible light communications (VLC) that uses a camera as the receiver. The main benefit of the paradigm is that there is already a pervasive existence of cameras in our daily life; mobile devices, laptops, cars, or, more recently, glasses (Google glass), all have built-in cameras nowadays, and thus it is possible to receive a small amount of information from LED lights, which are also installed pervasively in the environment for illumination, from virtually every IT device, creating a new communication infrastructure. The above two papers are some of the simpler implementations of camera communications. We encourage you to take additional considerations to make the system suitable for a wider range of scenarios. Note that since in lab 1 you have implemented a similar system, teams that select this topic have to implement additional functionalities for the final project.

A few possible topics:

1) Create a modulation format that can be received by cameras with different settings, that works in different lighting conditions (since the camera would be taking the picture with a longer shutter time), and that works when the

transmitting light and the receiving camera have different frame rates (synchronization issues). You can also try to include designs that could boost the data rate (such as utilizing the color).

2) LED-array-to-camera communications: Using a large number of LEDs to transmit at the same time to boost the data rate. We do have an arduino-compatible LED array – peggy2 board that you can use as a transmitter. Let us know if you'd like to use it for your project.
   http://www.winlab.rutgers.edu/~aashok/visualmimo/aashok_secon2011.pdf
   http://www.winlab.rutgers.edu/~aashok/visualmimo/mvarga_mobisys11.pdf
   Peggy2: http://wiki.evilmadscientist.com/Peggy_2
3) Vehicle-to-Vehicle CamCom: building on top of your lab 1, instead of using a small LED, we can lend you a set of hardware components that would allow you to modulate a real car or scooter taillight module. Then the goal is to be able to show that when the scooter/car are moving, a camera that takes images of the transmitting taillight can receive messages with high reliability. This CamCom system can enable a large number of safety applications, since the car would be able to report its current speed, location, steering angle, etc., via the taillight to surrounding vehicles.

6. GEMV^2: Geometry-based Efficient propagation Model for V2V communication (http://vehicle2x.net)

GEMV^2 is a propagation model for vehicle-to-vehicle communications. The propagation model takes into consideration many different aspects, including free-space path loss, shadowing caused by other cars or surrounding buildings, as well as small-scale fading caused by the environment and the velocity of the vehicle. The author of the model has an implementation of the model in MATLAB (which is quite cool; see the screenshot of its integration with Google Earth on the website). Michael is familiar with the author of the model so it is possible to directly ask him questions via e-mails. ☺

A few possible topic to utilize GEMV^2 MATLAB implementation:

1) Implementing the propagation model for vehicle-to-vehicle VLC.

The Vehicle-to-Vehicle VLC propagation model is slightly different from the RF propagation model; the most important two aspects are: (1) there is only a link if the transmitter and the receiver have line-of-sight (2) free-space path loss depends not only on the distance between the transmitter and the receiver, but also the respective angle between them. For this topic, you need to modify the original MATLAB code to incorporate this new VLC propagation model. Then, we would ask you to evaluate the performance of this new VLC-based vehicular network with realistic traffic traces generated by SUMO, a traffic simulator (the current implementation already supports that).

Vehicle-to-vehicle VLC propagation model:

http://www.csie.ntu.edu.tw/~hsinmu/wiki/_media/paper/vnc2013.pdf

SUMO: http://sumo-sim.org

2) Modify the model to take human body shadowing into consideration.

Scooter is very popular in Taiwan as well as several other Asian countries. Recent research results show that human bodies of the scooter driver and the passenger can create shadowing effect which attenuates the signal by up to 18 dB, when it blocks the line-of-sight. For this topic, you need to modify the original MATLAB code to consider this effect in the received power calculation.

Then, we would also ask you to evaluate the performance of the vehicular network with realistic traffic traces.

3) If you feel that you are already familiar with NS-2, you can also choose to implement GEMV^2 entirely from scratch in NS-2 and perform simple evaluation to show that it actually works.