Cellular Networks and Mobile Computing
COMS 6998-11, Fall 2012

Instructor: Li Erran Li
(lierranli@cs.columbia.edu)

http://www.cs.columbia.edu/~lierranli/coms6998-11/

9/4/2012: Class Introduction

Outline

- Introduction
- Course content
- Course goals and structure
- Example projects
- Programming environment setup

Introduction

- Researcher at Bell Labs, Alcatel-Lucent
- Ph.D. from Dept. of CS, Cornell, 2001
- Research interest: cellular networks, mobile computing, cloud computing
- Research Goal: improve our mobile user experience through innovation in cellular network architecture, network services, and mobile cloud computing

Experiences

- Relevant working experiences
  - Cellular networks: monitoring and trouble shooting
  - Mobile computing: mobile cloud computing
  - Cloud computing: scaling out enterprise applications, cloud-based video proxy, policy-aware enterprise application cloud extension

- Professional Activities
  - ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design (CellNet), August 2012
  - ACM MobiSys Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond (MCS), June 2010
  - DIMACS Workshop on Systems and Networking Advances in Cloud Computing, Dec, 2011

Introduction (Cont'd)

- Current research projects:
  - Eikon: A mobile smartphone performance virtualization architecture
  - mCloud: mobile cloud computing (ACM Mobisys MCS workshop'12)
  - Software-defined cellular networks (Euro SDN workshop'12)
  - LAWN: scaling up cellular networks using a large number of antennas (ACM MobiCom'12)

Who Are you?

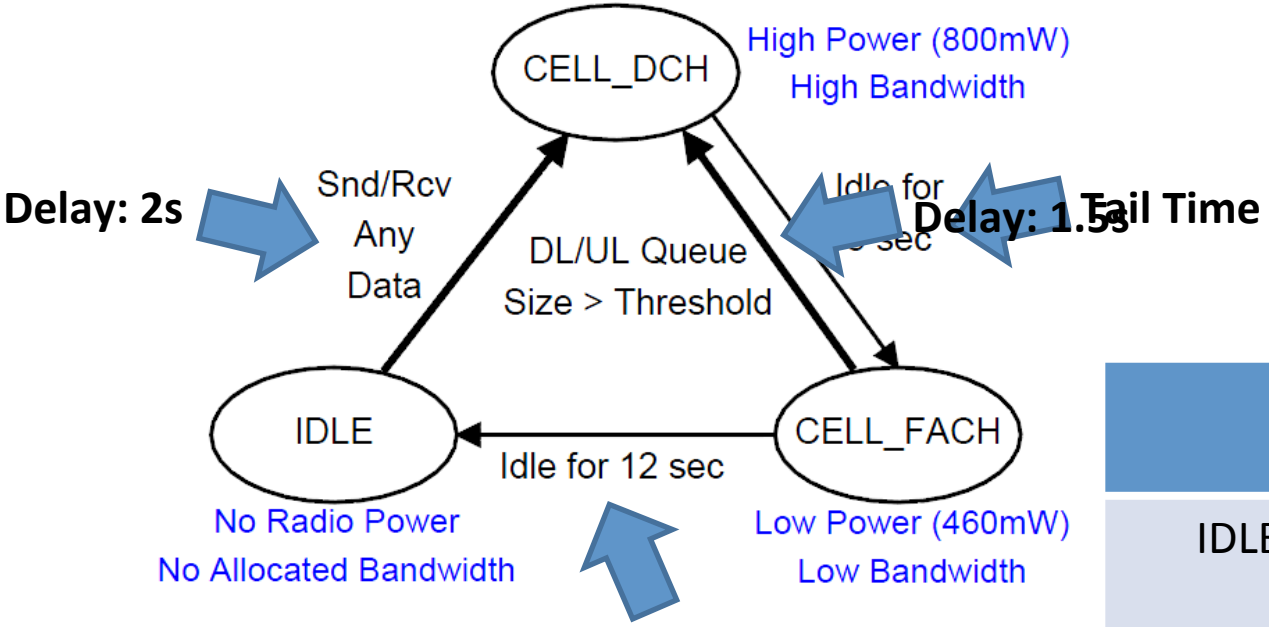
- Please briefly introduce yourself
  - Name
  - Program and year at Columbia
  - What do you want to learn from this course?

Course Content

- Why study cellular networks and mobile computing together?
  - Mobile apps with no knowledge of cellular networks can perform poorly
    - Pandora consumes 46% radio energy on periodic transfers of 0.2% received user data
  - Cellular networks with no knowledge of mobile apps can perform poorly, e.g. poor traffic planning, high latency for delay sensitive traffic

Example in Detail: The RRC State Machine for UMTS Network

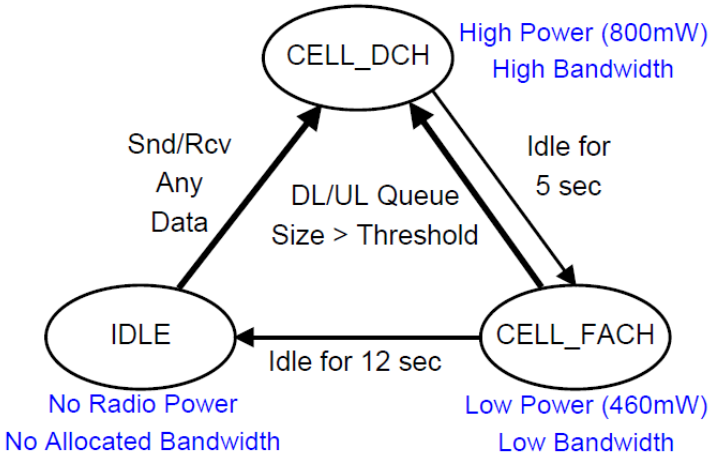
- State promotions have **promotion delay**
- State demotions incur **tail times**

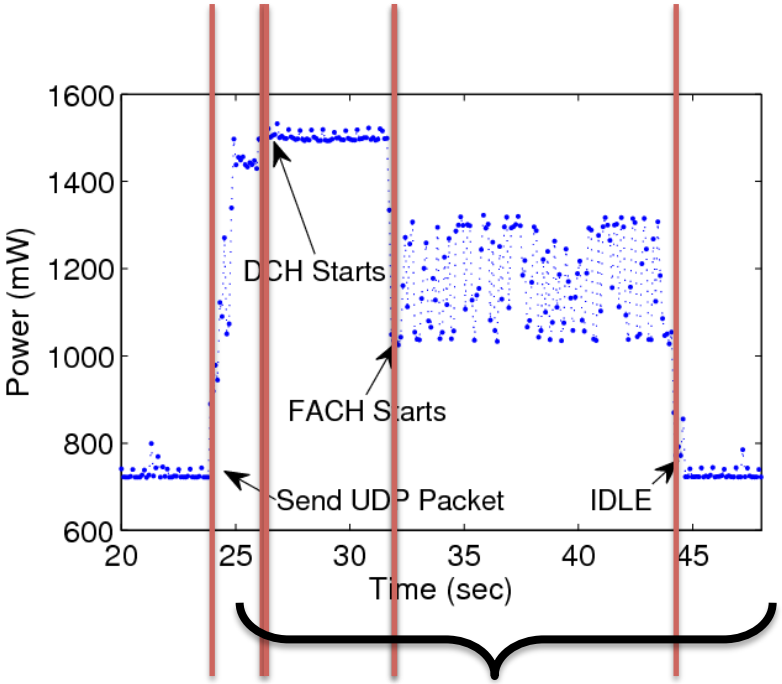


	Channel	Radio Power
IDLE	Not allocated	Almost zero
CELL_FACH	Shared, Low Speed	Low
CELL_DCH	Dedicated, High Speed	High

Courtesy: Feng Qian

Example in Detail: RRC State Machine for a Large Commercial 3G Network





Promo Delay: 2 sec DCH Tail: 5 sec FACH Tail: 12 sec

Tail Time

Waiting inactivity timers to expire

DCH: High Power State (high throughput and power consumption)

FACH: Low Power State (low throughput and power consumption)

IDLE: No radio resource allocated

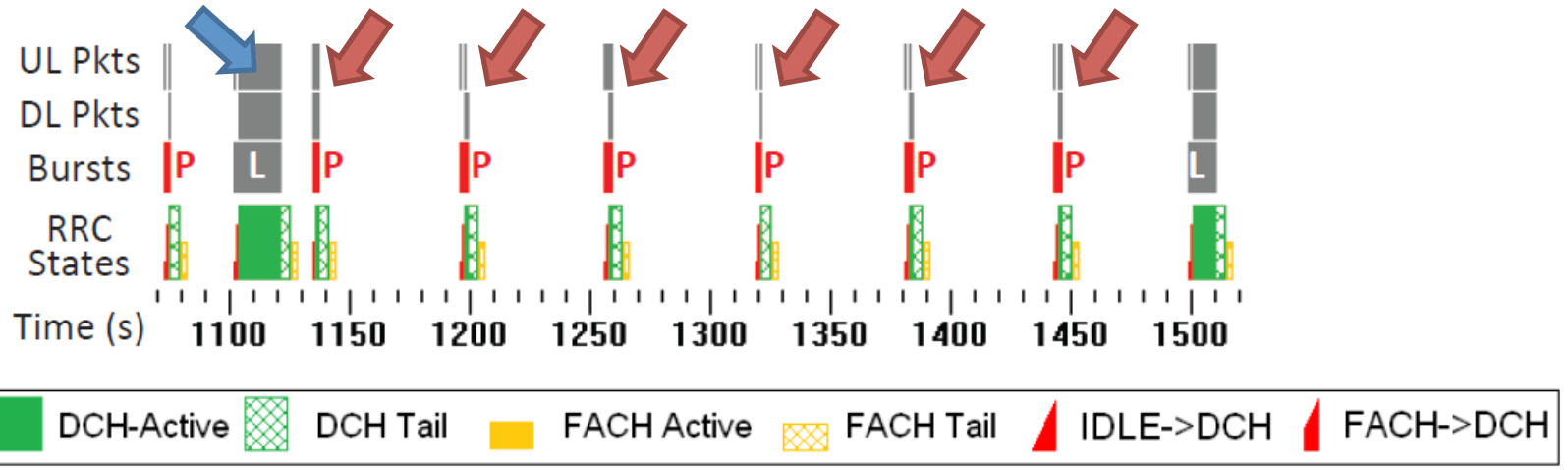
Courtesy: Feng Qian

Example in Detail: Pandora Music

Pandora profiling results (Trace len: 1.45 hours)

Burst type	Payloads	Energy		DCH	
		LB	UB	LB	UB
LARGE_BURST	96.4%	35.6%	35.9%	42.4%	42.5%
APP_PERIOD	0.2%	45.9%	46.7%	40.4%	40.9%
APP	3.2%	12.8%	13.4%	12.4%	12.8%
TCP_CONTROL	0.0%	1.2%	1.6%	1.1%	1.5%
TCP_LOSS_RECOVER	0.2%	0.2%	0.6%	0.3%	0.7%
NON_TARGET	0.0%	1.8%	1.8%	1.7%	1.7%
Total	23.6 MB	846 J		895 sec	

Problem: High resource overhead of periodic audience measurements (every 1 min)
Recommendation: Delay transfers and batch them with delay-sensitive transfers



Courtesy: Feng Qian

Example in Detail: Feedback from Pandora

AT&T's analysis of the Pandora application gave us a much better view of how Pandora interacts with low-level cellular network resources. Now that we better understand these interactions, we can optimize our application to make more efficient use of these resources. In fact, we'd like to incorporate AT&T's profiling tool as part of our normal ongoing testing.

Tom Conrad, CTO of PANDORA®

Courtesy: Feng Qian

Course Content (Cont'd)

- This course has three themes revolving around improving mobile user experience
  - Understand current cellular networks and their interaction with mobile apps through measurements
  - Improve the interplay of cellular networks and mobile computing through new cellular network services (e.g. proxy, caching), cellular aware mobile app design, redesign of cellular networks
  - Improve mobile apps through cloud computing such as novel cloud platform services (e.g. iCloud, Amazon Silk Split Browser, push notification server)

Course Goals and Structure

- Basics: brief overview of cellular networks and mobile OS and development platforms
- Recent literature: review recent research on cellular network measurements, and mobile computing
  - Paper presentation, summary, and discussion
- Learn by doing: work on a research project

Basics

- Overview of cellular networks
  - UMTS(3G) and LTE: air interface, architecture, mobility management
- Mobile OS and development platform
  - iOS development platform: Xcode, model-view-controller programming model, Objective-C features, iCloud
  - Android programming
- Cloud computing
  - Google AppEngine
  - Amazon EC2

Recent Literature

- Will read about 22 papers that identify or address challenges in cellular networks and mobile computing
- Papers covered will be in networking, systems and security; topics include
  - Understanding the interplay of cellular networks and mobile computing through measurements
  - Cellular aware mobile application design
  - Power models of mobile devices
  - Smartphone virtualization and storage
  - OS support for energy and sensor management
  - Mobile cloud computing
  - Mobile security and privacy

Recent Literature (Cont'd)

- Your duties:
  - Read all assigned papers before class
  - Participate in class discussions
  - Present and summarize 1 or 2 papers

Research Project

- Topic
  - Choose from a list of topics
  - Come up with your own topic
  - Must be related to cellular networks or mobile computing
  - Must contain some research element
- Teams of 2 to 3 students
- Final deliverables
  - Project report (research paper format, 10 to 12 pages)
  - Project presentation and demo

Research Project (Cont'd)

- Precisely define the project
- Understand related work
- Propose novel techniques or systems
  - Creativity will be evaluated
- System implementation
  - Client side: iOS or Android
  - Server side: Google AppEngine or Amazon EC2
  - Networking component: measurement, modeling

Research Project (Cont'd)

- Evaluate your solution, e.g. performance, scalability
  - Thoroughness will be evaluated
- Write up and present your projects
  - Evaluated using professional paper review criterions

- Project timelines
  - Sept. 18: Form final project team
  - Oct. 2: project description
  - Oct. 23: preliminary project report
  - Dec. 4: final presentation and demo
  - Dec. 11: final project report
- I will meet with you regularly

Grading

- Project reports: 50%
- Three programming assignments: 30%
- Paper presentation and summary: 10%
- Class discussion participation: 10%

Class Resources

- Web page: schedule, project timelines, list of potential projects, etc

- For any questions or concerns: email me at lierranli@columbia.edu

Example projects

- Ideal project criterions
  - Solves a real problem in cellular networks and mobile computing
  - Has a research component, e.g. scalable system design, novel inference algorithm of cellular network properties
  - Real implementation at client side running iOS or Android, and at server side using public cloud platforms such as Google AppEngine or Amazon EC2

Example project 1: iConnect

- Goal: build an app that makes file sharing easy among iOS devices
- Research: multi-hop issue and low power usage
- Implementation: Bonjour, file system API and the network API
  - Bonjour service discovers devices in the same wireless network
  - The file system API is used to select or store files
  - The network API sets up network connection

Example project 2: EventDroid

- Goal: build a general event management and data sharing app
- Research: scalable server and client system design
- Implementation:
  - C2DM is used to notify clients efficiently (avoids polling)
  - Google Cloud SQL to manage events

Programming environment setup

- Client side: iOS
  - Install Xcode 4: http://developer.apple.com/xcode
  - Learning Objective C and iOS development : http://developer.apple.com/devcenter/ios/index.action
  - Stanford iPhone development course(on iTunes): http://www.stanford.edu/class/cs193p/cgi-bin/drupal/

Programming environment setup (Cont'd)

- Client side: Android
  - Install Eclipse: http://www.eclipse.org/downloads/
  - Install Android SDK: http://developer.android.com/sdk/index.html
  - Android programming resources: http://developer.android.com/index.html
  - Stanford course: http://www.stanford.edu/class/cs193a/

Programming environment setup (Cont'd)

- Server side: Google AppEngine
  - Install: http://code.google.com/appengine/
  - Install plugin for Eclipse: http://code.google.com/appengine/downloads.html#Download_the_Google_Plugin_for_Eclipse

- Amazon EC2: http://aws.amazon.com/ec2/
  - Free usage tier: http://aws.amazon.com/free/

Questions?