



A Quick Look At

6.02 Intro to EECS II

Hari Balakrishnan

Dina Katabi

Michael Perrott

Charlie Sodini

Vladimir Stojanovic

Chris Terman

Jacob White

6.02 Mission

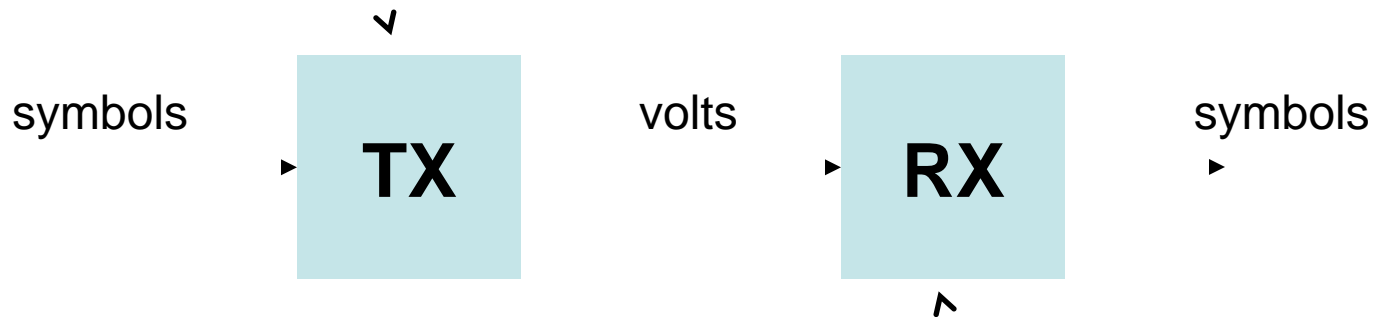
- Expose students to real engineering
 - Leverage natural curiosity about how things work
 - Finding and using appropriate analysis approaches
 - Issues in making big, long-lived engineering artifacts
- Hands-on experience with important EECS topics
 - Signals and systems
 - Information theory
 - Algorithms
 - Systems architecture: theory and practice
- Approach: vertical slice through the engineering of an interesting system: **digital communication networks**
 - Lots of contact hours (2 lec, 2 recitations each week)
 - Time to think things through in a supportive environment: asynchronous labs can be done independently but course homeroom offers daily help, support, interaction with peers
 - Incorporate design activities in weekly labs





Lab 1: Getting Started

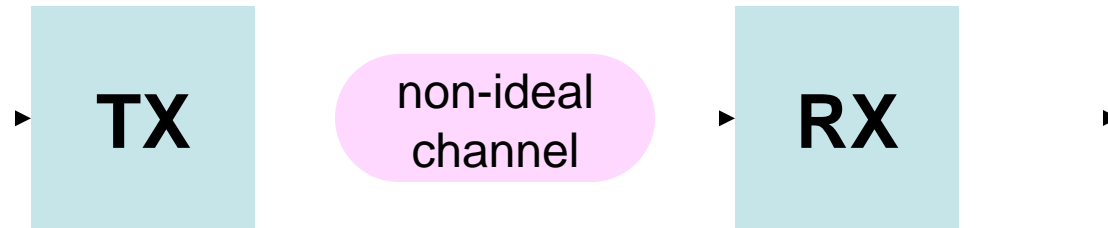
Digital transmitter using PAM to encode symbols to be sent over a “perfect” channel (just attenuation, delay)



Think about how to recover clock and data from received signal: use edges to encode clock, levels to transmit data.

Design question: how to ensure enough edges? Hint: TX is allowed to recode symbols (eg, 8b/10b recoding in PCI-E, SATA)

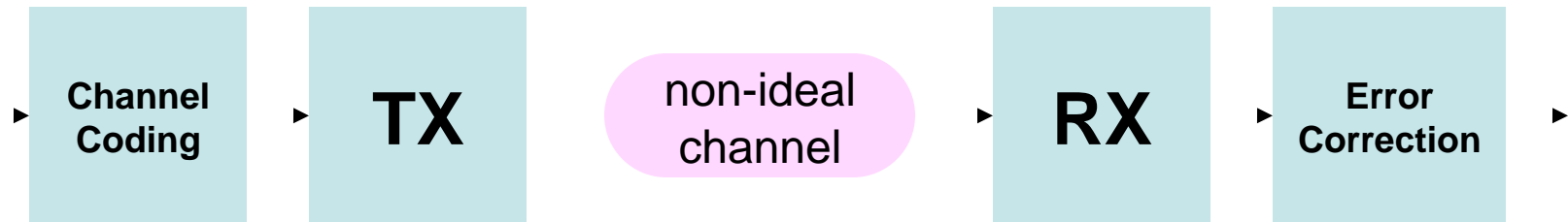
Labs 2 & 3: Reality strikes back



- LTI channel models, unit-sample response
- Inter-symbol interference, eye diagrams
- Complications when there's noise (AGWN)
- Relating noise PDF to bit-error rate
- Conversion back to symbols: what's the best slicing level?

Design question: knowing the unit-sample response, what can the transmitter do to reduce ISI at the receiver?

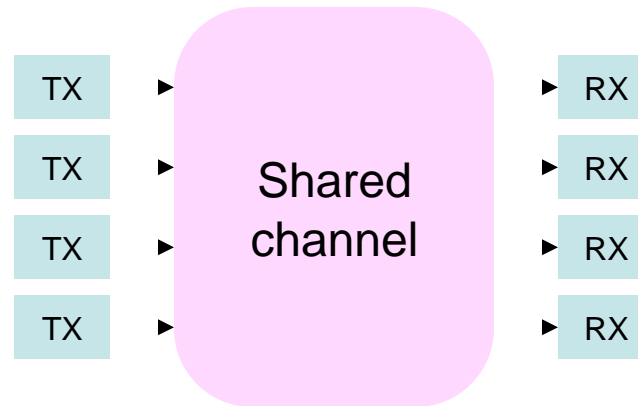
Labs 4&5: Dealing with errors



- Thinking about bit errors: Hamming distance
- Detecting errors (parity, checksums, CRCs)
- Correcting errors (“triangulation” using multiple syndrome bits)
- “Continuous” coding with convolutional codes
- Implement decoder using Viterbi algorithm

Design question: how to deal with error bursts?

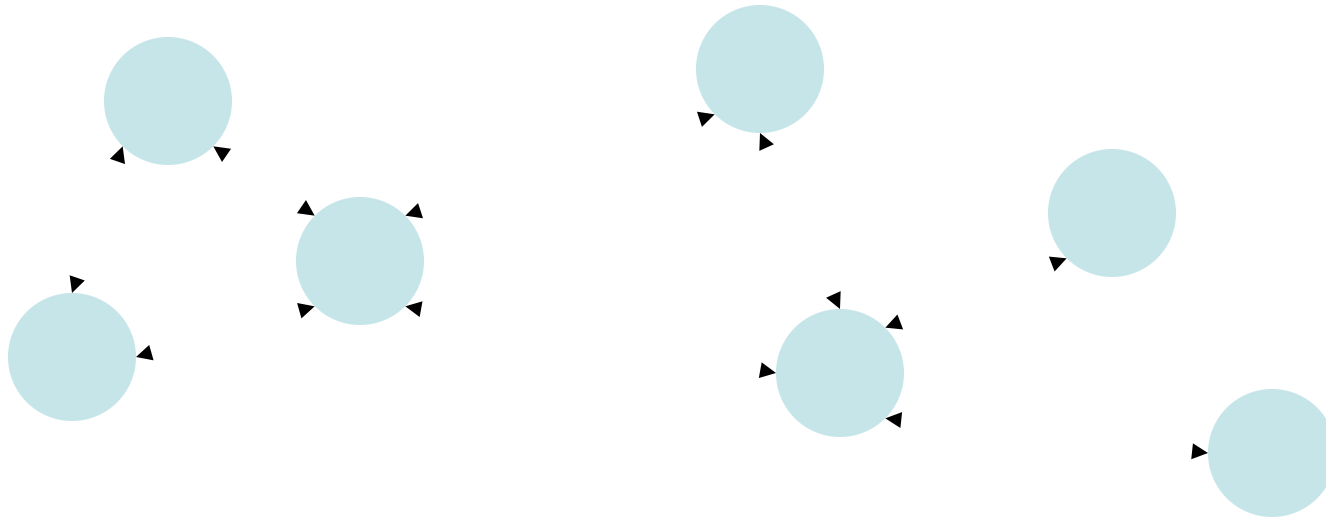
Labs 6, 7 & 8: Sharing



- Channel sharing using orthogonal basis functions
- Sinusoidal basis functions & LTI channels
- Thinking in the frequency domain: the Fourier transform
- Multiplication!
 - Amplitude modulation (time domain)
 - Filtering (frequency domain)
- Frequency-domain multiplexing
- Time-domain multiplexing

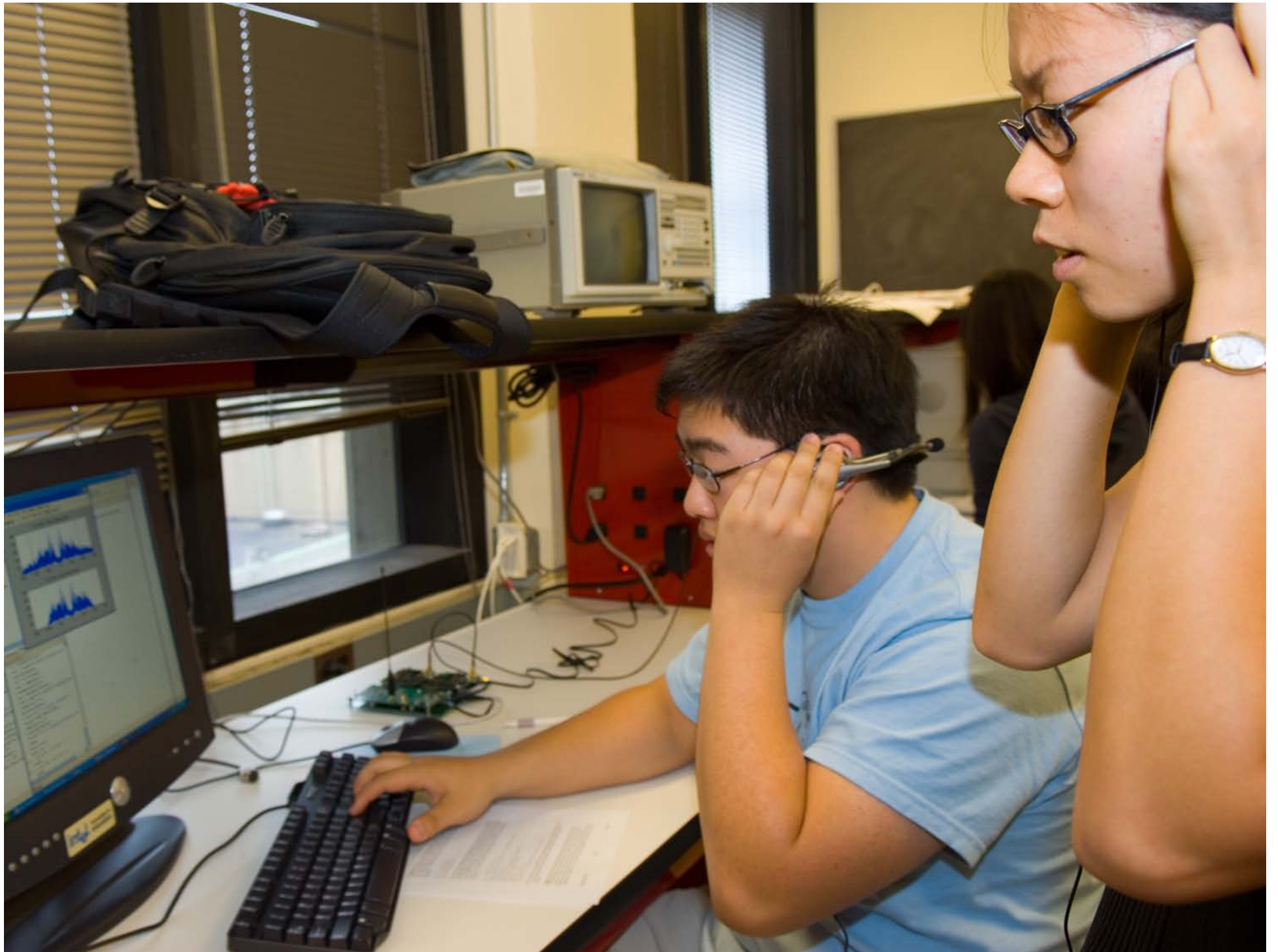
Design question: how to implement contention-based time-domain multiplexing?

Labs 9 & 10: Multi-hop networks



- Switching principles: circuits vs. packets
- Network layer: addressing and shortest-path routing
- Transport layer: reliable data delivery on top of a best-efforts network layer

Design question: how to deal with link failure?



Our experimental apparatus

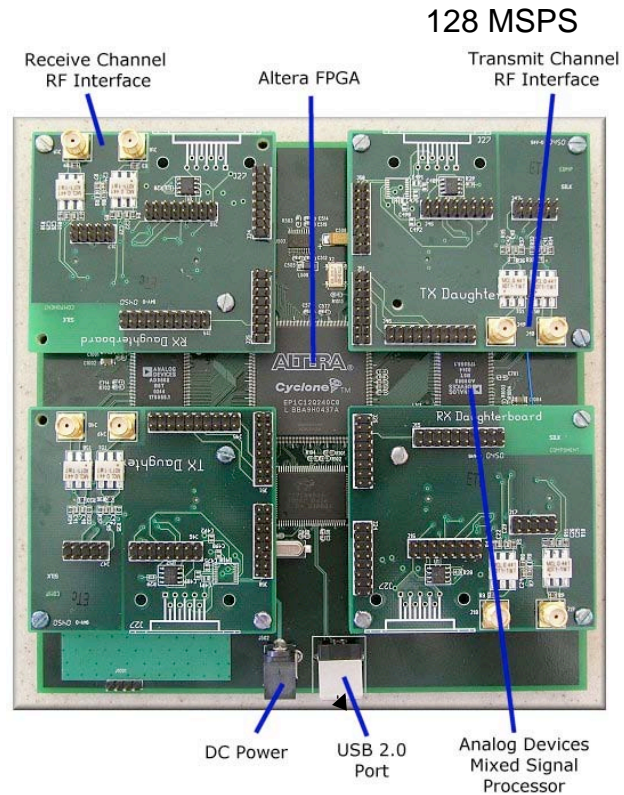
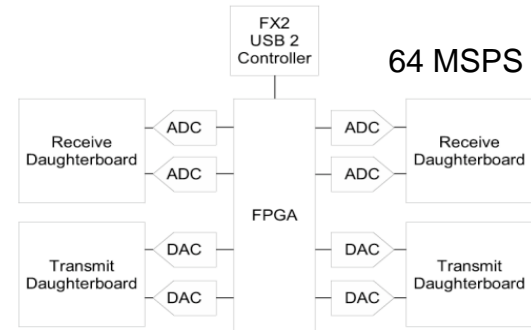


Analog Headset



Workstation:
Python/Numpy
Streaming tools

USB cable



USRP board