Cognitive Radio Rendezvous
Craig Guterman, Kyle Soska
Advisors: R. Yates, I. Seskar, K. Le, P. Maddala

Motivation
- Uncoordinated Cognitive Radios need to agree upon a common channel and synchronize in time
- Goal: Design Rendezvous algorithm to synchronize two radios quickly as possible

System Model
- Measure interference of channel with 64-pt FFT
- Determine which channels are available
  - Channels: $C = \{C_1, \ldots, C_{16}\}$
  - Available Channels Radio 1: $C_1 \subseteq C$
  - Available Channels Radio 2: $C_2 \subseteq C$
  - $C_1 \cap C_2 \neq \emptyset$
- The channel hopping sequence follows:
  - $C_1(t) \in C_1$
  - $C_2(t) \in C_2$
  - Transmitting: $m_t(t) = 1$
  - Receiving: $m_r(t) = 0$
- Rendezvous Time:
  - $R = \min \{t \geq 0 \mid C_1(t) = C_2(t), m_t(t) = m_r(t)\}$

Network Algorithm
- Use a slotted networking protocol
- Data Stream 1: Sends Spectrum Data to Host
- Data Stream 2: Sends and Receives data packets to Host

Rendezvous Process
- Develop Rendezvous Process
  - Interference is Asymmetric
  - Time Asynchronous
  - Use Jump Stay Based Hopping Sequence
- Process begins by listening for a beacon
- Transmits Beacon after random back-off
- Continues to listen for response before switching channels
  - Message not recognized if on different channel
  - Algorithm continues until successful handshake
- Rendezvous not guaranteed even if on same channel

Results
- Parameters Varied:
  - Size of Window
  - Random back-off time interval
- Measure:
  - Average number of attempts to rendezvous
  - Average time To rendezvous (Avg. (TTR))
- Results:
  - 4ms Window Size Minimizes Avg. (TTR)
  - Small Backoff reduces TTR