DARPA Spectrum Collaboration Challenge (SC2)

Team GatorWings

UNIVERSITY *of* FLORIDA

Our Team - Faculty

Tan Wong

- ➔ Team Lead
- ➔ PHY & MAC
- ➔ Decision Engine

John Shea

- ➔ Team Co-Lead
- ➔ Link & Network Layers
- ➔ Pocket Scheduler/Decision Engine

Our Team – Graduate & Undergraduate Students

David Greene

FPGA
Network Layer
IDE & Tools

Tyler Ward

Physical Layer
Link Layer
Control Plane

Marco Menendez

CIL Development
Workflow
Optimization

Caleb Bowyer

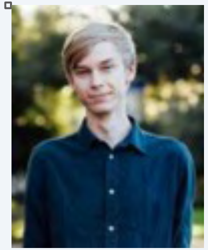
Machine
Learning

Shiming Deng[1]

Analysis &
Visualization Tools

Quan Pham[2]

Channel Emulation



Josh Agarth[3]

CIL Development

[1]Phases 1 & 2: Now with L3Harris

[2]Phases 1 & 2: Now in UF MS Program

[3]Phase 1: Now with Lockheed-Martin

SC2 — DARPA's 5th Grand Challenge

"The first-of-its-kind **collaborative machine-learning competition** to overcome scarcity in the radio frequency (RF) spectrum"

— *spectrumcollaborationchallenge.com*

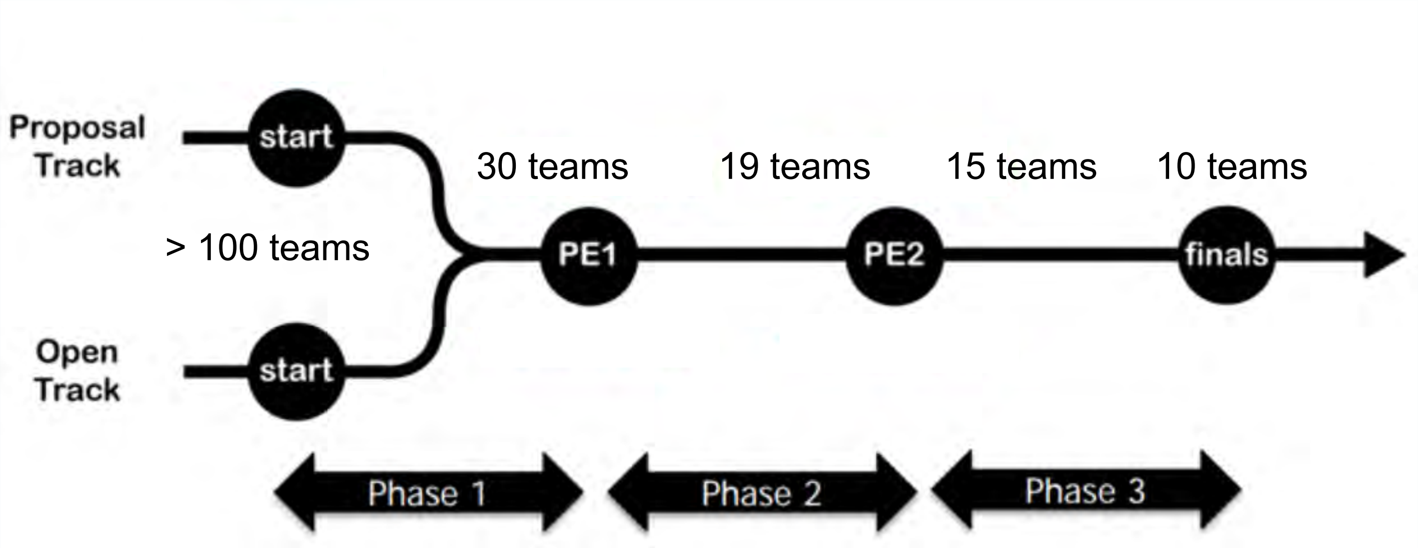
Conventional approach to spectrum management

- Spectrum is assigned by *humans* (FCC & NTIA)
- Spectrum allocations are typically permanent or last for years
- Allocated spectrum is often unused (white space)

SC2 Design Challenge

- Develop new approaches to spectrum management by **autonomous**, **intelligent** agents (take humans out of the loop)
- Spectrum management should be **distributed** and **collaborative** across diverse networks
- Spectrum management should dynamic
  - Time scales of **seconds** instead of **days** or **years**

SC2 – Competition Structure



* Figure modified from DARPA's "Competitor Information Day Slides"

	Phase 1 / PE1	Phase 2 / PE2	Phase 3 / SCE
Network size (# of radios)	5	10	10
Match size (# of teams)	3	5	5
Tournament date	Dec. 2017	Dec. 2018	Oct. 2019

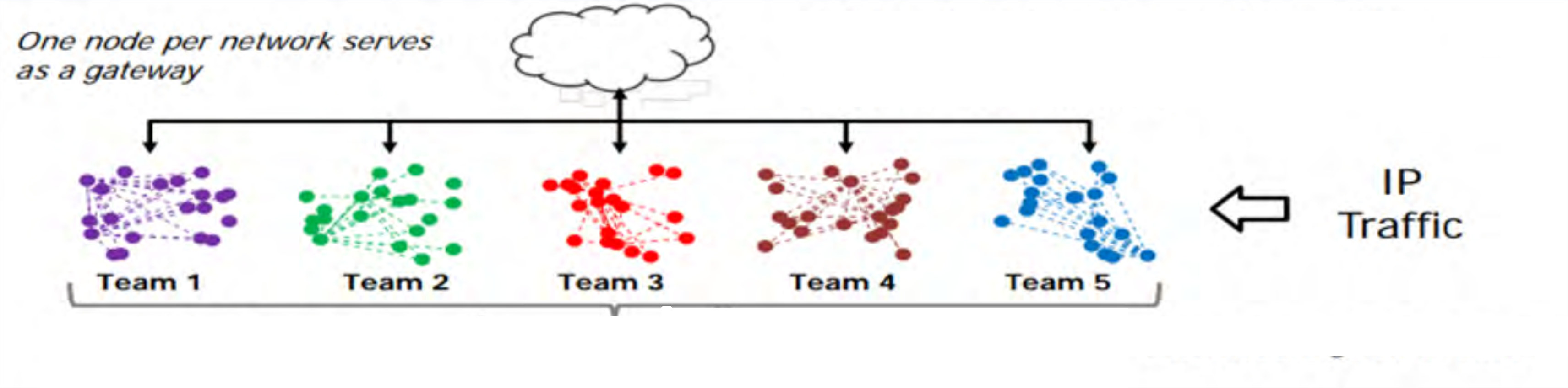
SC2 Championship Event, Oct. 2019

- 10 teams
- Held live at Mobile World Congress, Los Angeles

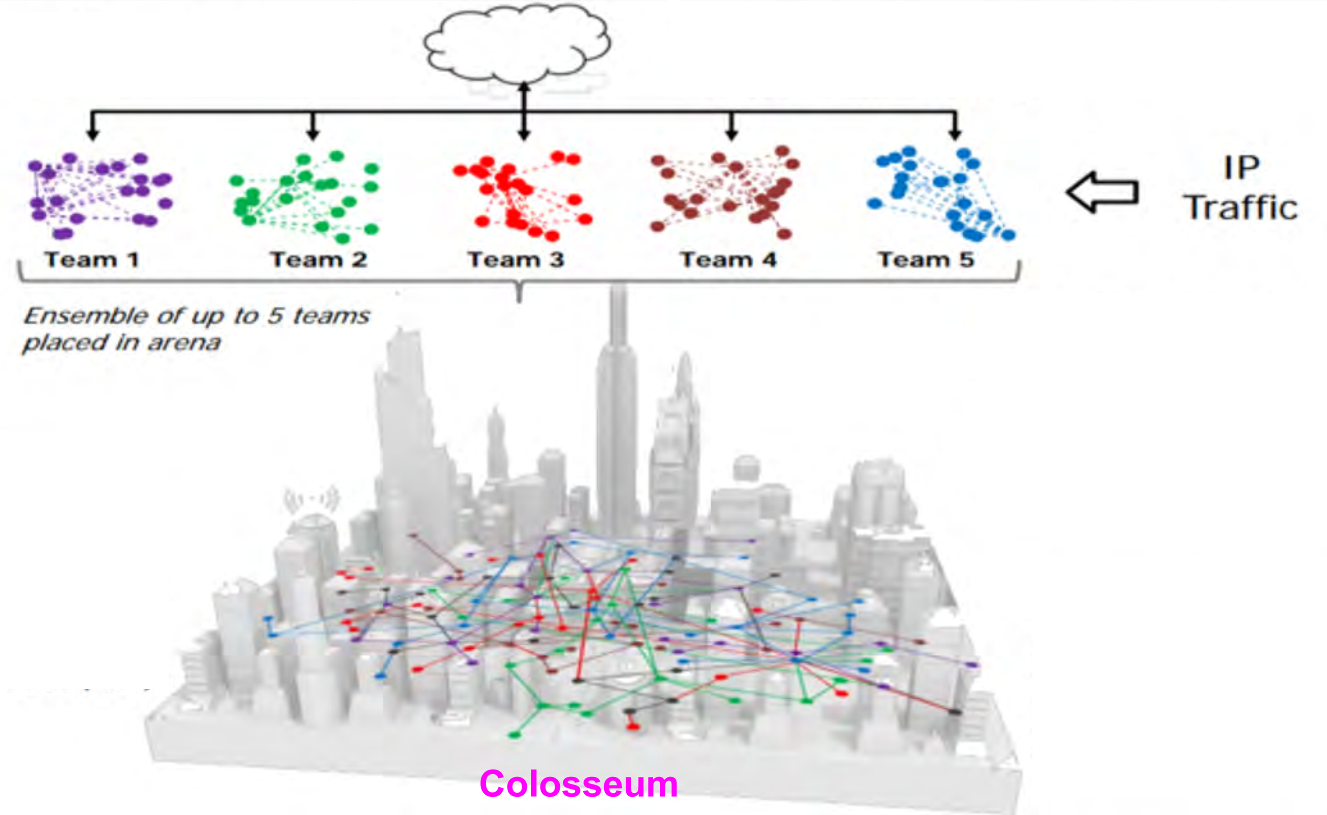
- **Team GatorWings was only team in final round with students**

SC2 - Tournament Match

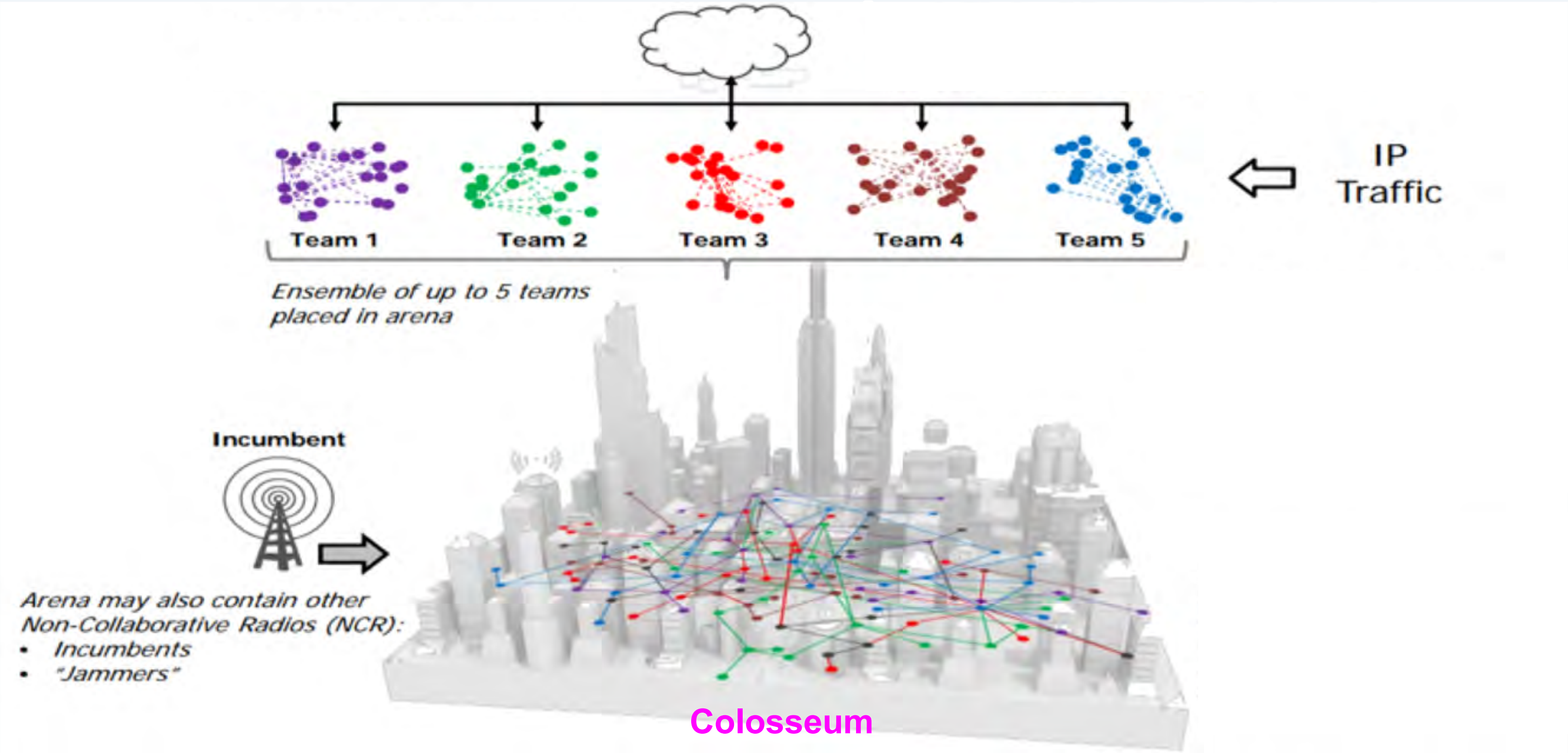
* Figure from DARPA's "Competitor Information Day Slides"



SC2 - Tournament Match



SC2 - Tournament Match

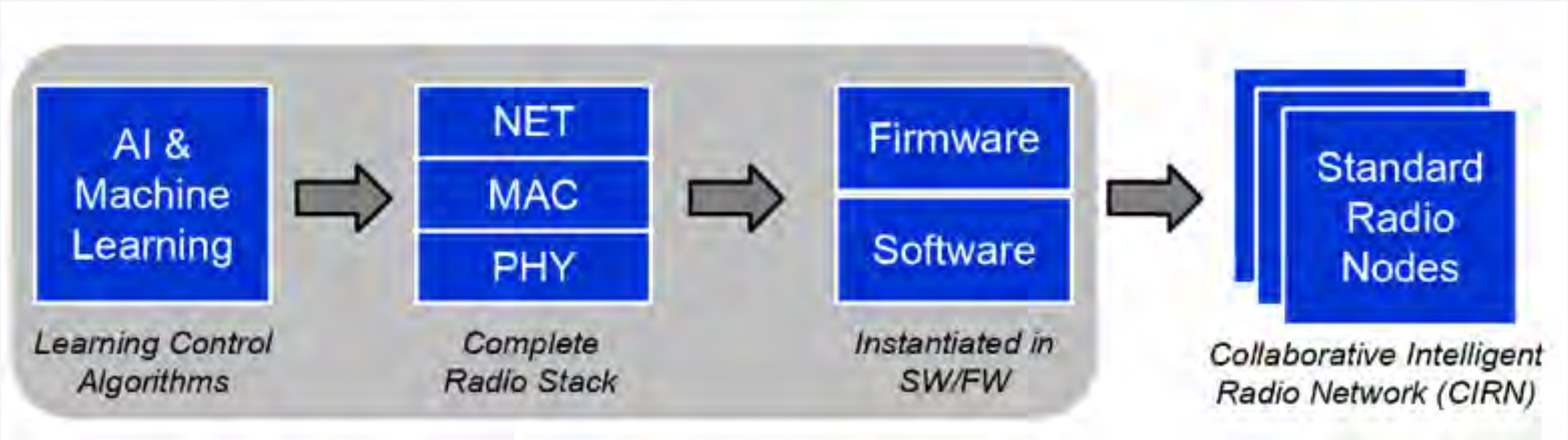



COLOSSEUM

- Massive Channel Emulator (MCHEM)
  - 256 x 256 100 MHz real-time RF channel emulation
  - Can run multiple simultaneous traffic and RF scenarios
- 128 Standard Radio Nodes
  (Ettus X310 USRP and Server)
  - 2 TX and 2 RX independent RF chains
    — can be Configured as 2x2 MIMO
  - Processing/computing resources:
    48-core CPU, GPU, FPGA

Match Scenario: Slice of Life

SC2 - Competitor’s Job



Strategy and Technical Approach

SC2 Problem:

- Dynamic spectrum access environment with 5 teams/networks of radios communicating in same frequency band
- Each team scores points by delivering traffic flows achieving certain QoS mandates (throughput, latency, hold time, etc.)

SC2 Problem:

- **A mixed cooperative/competitive game**

- Team's match score $= \begin{cases} \text{min score} & \text{if min score} \leq \text{scoring threshold} \\ \text{achieved score} & \text{if min score} > \text{scoring threshold} \end{cases}$

  where min score = minimum among all 5 team score

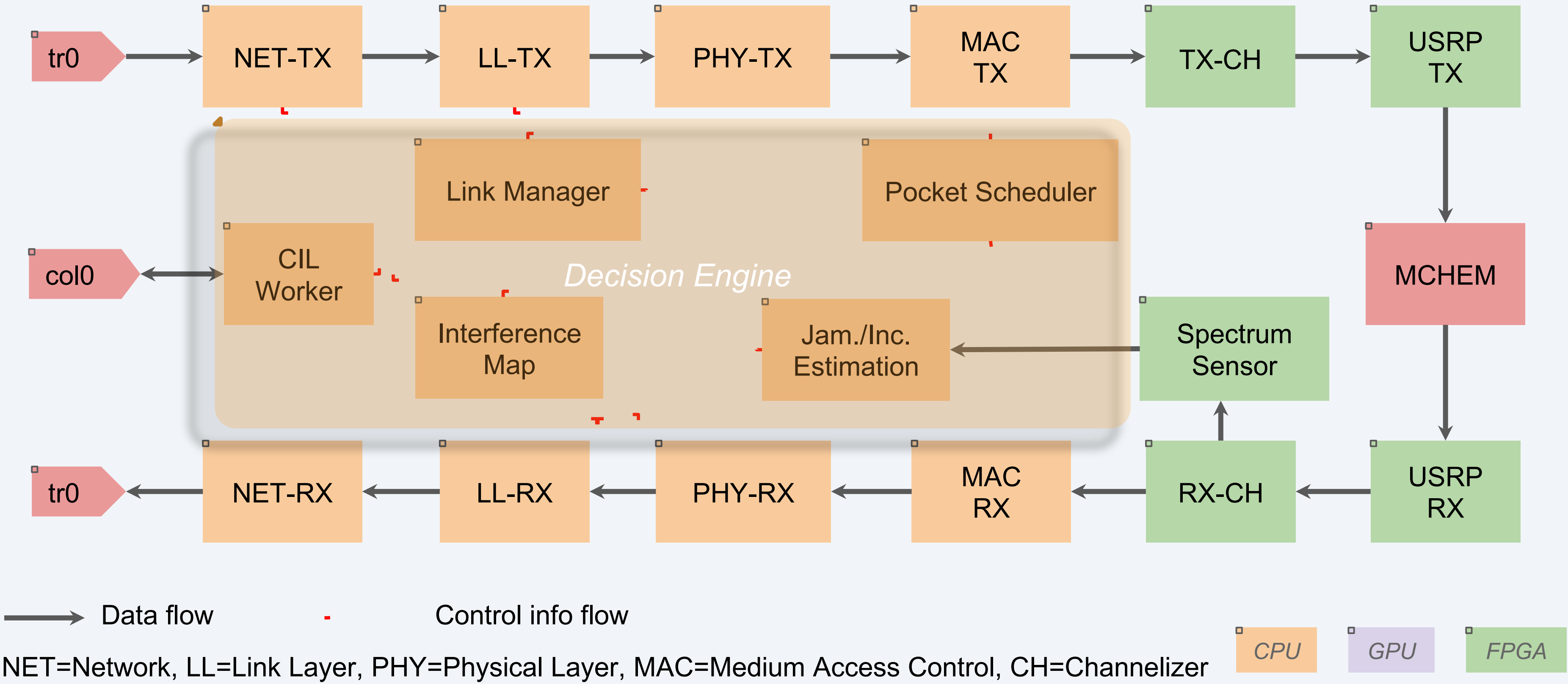
SC2 Problem:

- Adapt strategy in presence of **rich but incomplete information**:
  - No online scoring information, other than teams' estimates
  - Teams use CIL to report frequency use, radio locations, performance (score) estimates
    - Some CIL veracity checks on spectral use, scores
    - Teams do not have to report their true scores when their scores are above the threshold
  - Incumbents report channel usage, interference received and threshold, threshold violations
  - Spectrum sensing to estimate peer channel usage and detect jamming and active incumbents

Our Strategy - 1

- Design custom radio stack that is:
    - **Flexible/agile:** able to exploit opportunities in time, frequency & space (cooperative + competitive)
    - **Robust:** adaptive and capable of operating in presence of strong interference (competitive)

Overall Radio System



NET=Network, LL=Link Layer, PHY=Physical Layer, MAC=Medium Access Control, CH=Channelizer

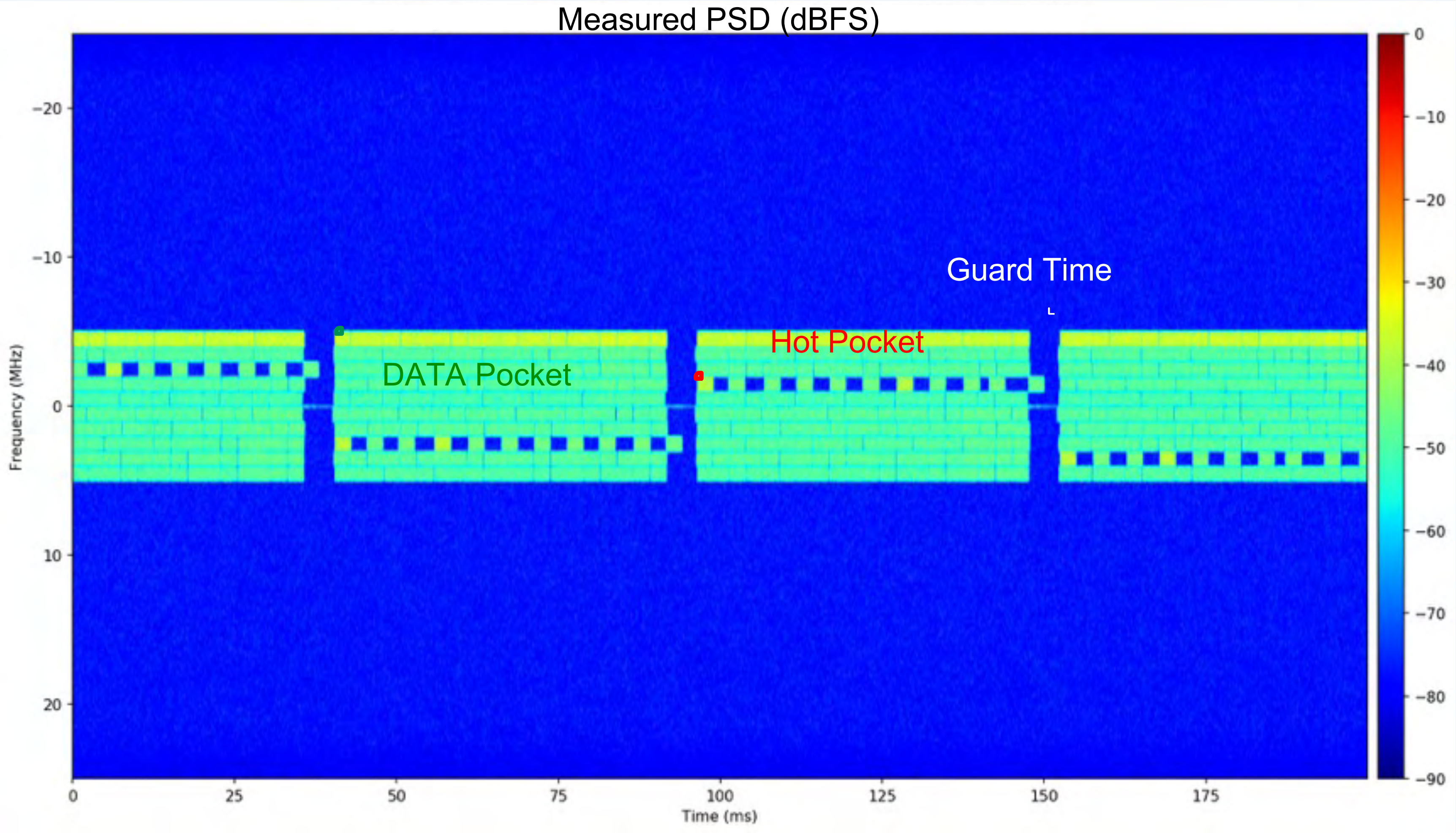
MAC: Time-Freq Pocket Structure

Time

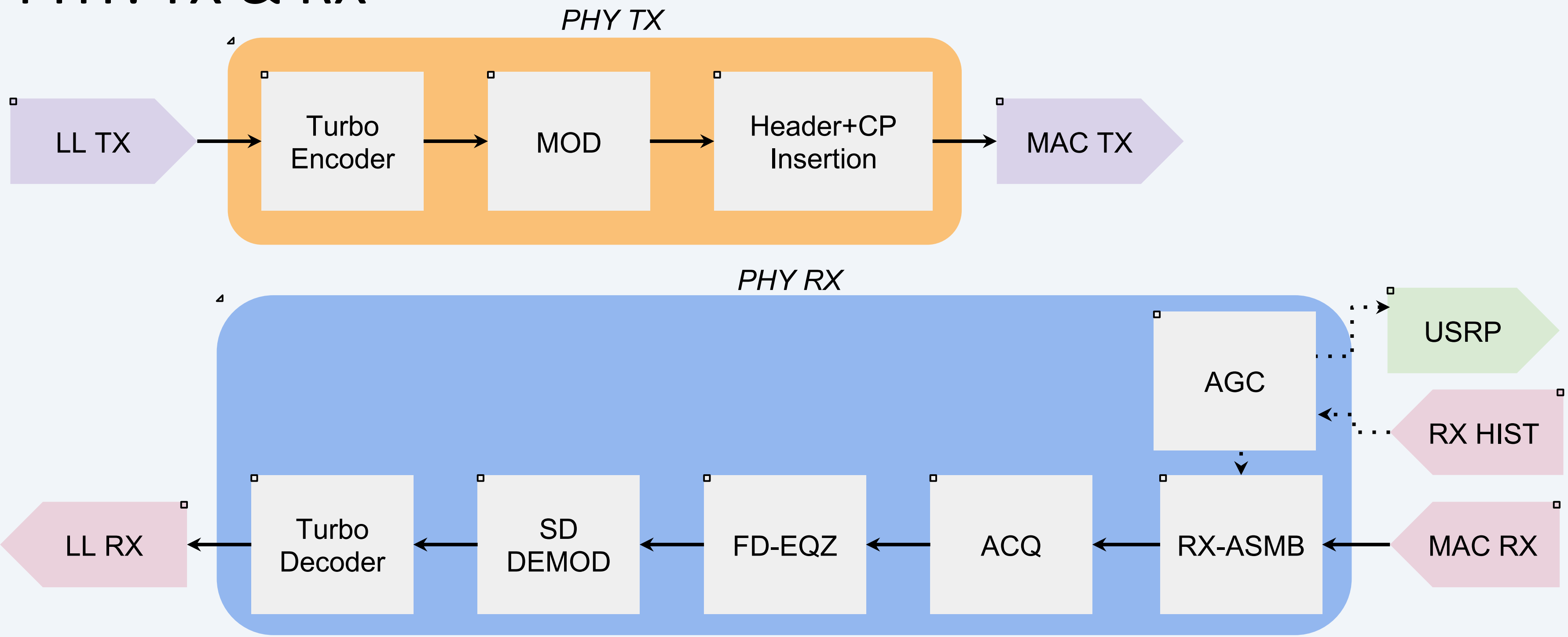
Freq / Channel

DATA	DATA	DATA	HOT POCKET	DATA
DATA	DATA	DATA	DATA	HOT POCKET
DATA	HOT POCKET	DATA	DATA	DATA
DATA	DATA	HOT POCKET	DATA	DATA
HOT POCKET	DATA	DATA	DATA	DATA

- 40MHz bandwidth channelized into 79 1MHz channels separated by 0.5MHz
- Dynamically choose a subset of non-overlapping channels to use
- Repeating frames of 10 time slots, each 56ms in duration
- Each pocket (time-freq slot) can deliver **ACKed data** from one SRC to **multiple** DSTs
- Randomized “hot pockets” for broadcasting network management info **and ACKs**
- Packetized PHY transmission
- Each radio can simultaneously **transmit and receive on multiple channels**


Measured PSD (dBFS)
Guard Time
Hot Pocket
DATA Pocket
Frequency (MHz)
−20
−10
0
10
20
0
25
50
75
100
125
150
175
Time (ms)
0
−10
−20
−30
−40
−50
−60
−70
−80
−90

PHY: TX & RX



- Turbo code: Incremental redundancy with base rate-5/6 code, rate-½ code for robust TX
- Adaptive modulation: QPSK, 16QAM, 64QAM
- FD-EQZ: CP insertion can be turned off; FD-EQZ does CFO correction, phase sync, RX beamforming

Radios Must Adapt and be Robust to Many Challenges:

Everything is adaptive

- PHY: Acquisition, Modulation, Coding, TX Power, RX Gain
- LL: Channels and Time Slots/Channel, Mapping of SRCs to Time Slots
- NET: Supported flows, admission control granularity down to individual files/bursts
- Other: Channels to jam

Spectrum Access Action by Decision Engine

- Decision engine attempts to maximize our team’s match score :
  - **which flows are transmitted**
  - **which channels are used and by which radios**
  - **which flows are sent in which pockets**
- Spectrum access action = **Pocket Schedule**
- Action space is huge!
  - 40 channels x 10 time slots = 400 pockets
  - As many as 100+ flows
  - 100^{400} possible pocket schedules!

Inputs to Decision Engine

- QoS mandates for our team’s flows
- Estimated number of achieved mandates and total mandates for our network
- Information on throughput per pocket expected between each SRC-DST pair
- Peer networks’ IDs (identified based on CIL message characteristics)
- Channels used by our network and by peer networks
- Estimated channel occupancies from our spectrum sensor (PSD measurements)
- Computed SINRs from our interference map (GPS and voxel info from CIL messages)
- Estimated achieved and total mandates from competitor networks (Performance info from CIL messages)

Decision Engine Design

- **No ML black box that can solve spectrum access problem**
- Decompose problem into smaller pieces:

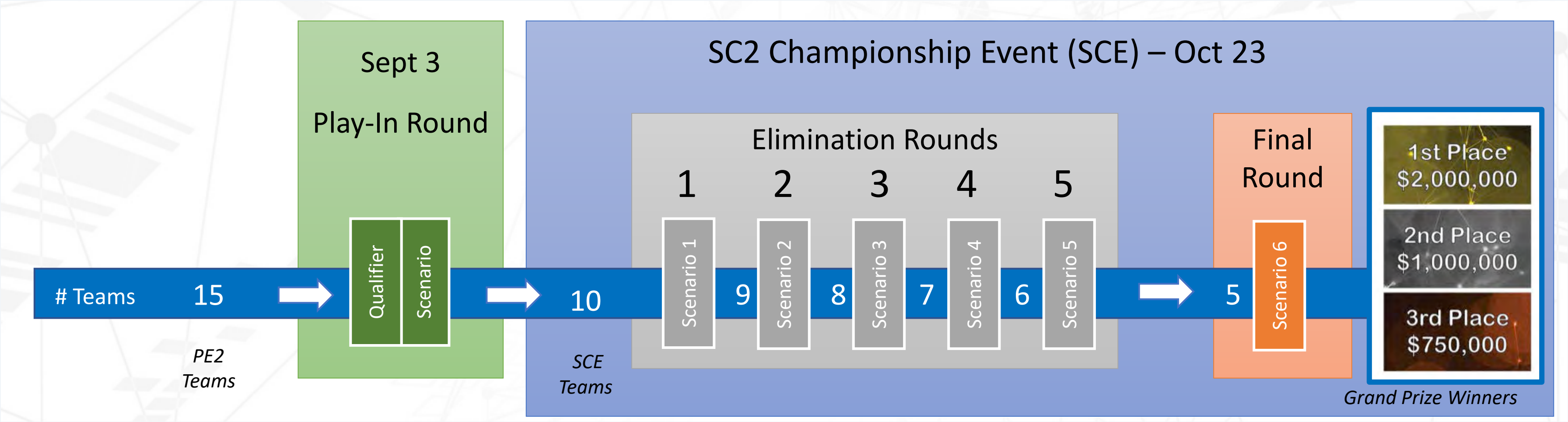
1. **Channel selection**
   - Determines target set of channels *C* to be used by our network
   - ML and expert system/control/optimization approaches
2. **Admission control**
   - $|C|$ determines number of pockets available
   - Estimates number of pockets needed to support each flow
   - Iterative process to determine set of flows to admit in order to maximize points scored
3. **Pocket schedule assignment**
   - Linear program to allocate number of pockets to satisfy latency requirements of all admitted flows
   - Greedy algorithm to assign pockets in each frame to satisfy mandates of all admitted flows
   - Maps to channels in *C* based on worst-case SINR over links of SRC-DST pairs in above assignments

Channel Selection by Reinforcement Learning

- Still need to reduce size of state space:
  - Construct “right” features from input parameters to maximize match score accounting for cooperative/competitive scoring function
  - Agent specialized for each peer network and each scenario
- Trained (SARSA) offline using data from Freeplay jobs
- Q-value function filled in by standard neural network with one hidden layer
- Policy used online
- Total number of channels from all peers by aggregating actions from all agents

- *Not used in SCE because of limited training data, rapidly changing peer strategies, and conflicting match/round/championship objectives*

SCE Competition Structure



Our Strategy - 2

- To make it more difficult in the SCE, elimination and final rounds have **conflicting objectives**
  - Elimination rounds: Maximize cumulative match scores over many matches (more cooperative)
  - Final round: Maximize match score in each match, i.e., rank as high as possible (more competitive)

- **Competitive > Cooperative in SCE:**
  - No info to identify whether in elimination or final round
  - Very difficult to have a strategy that works best in both because of conflicting objectives
  - Final round determines winner
  - Just need to survive elimination rounds
  - Elimination matches also favors competitive

Spectrum sharing strategy to win each individual match (more competitive)!

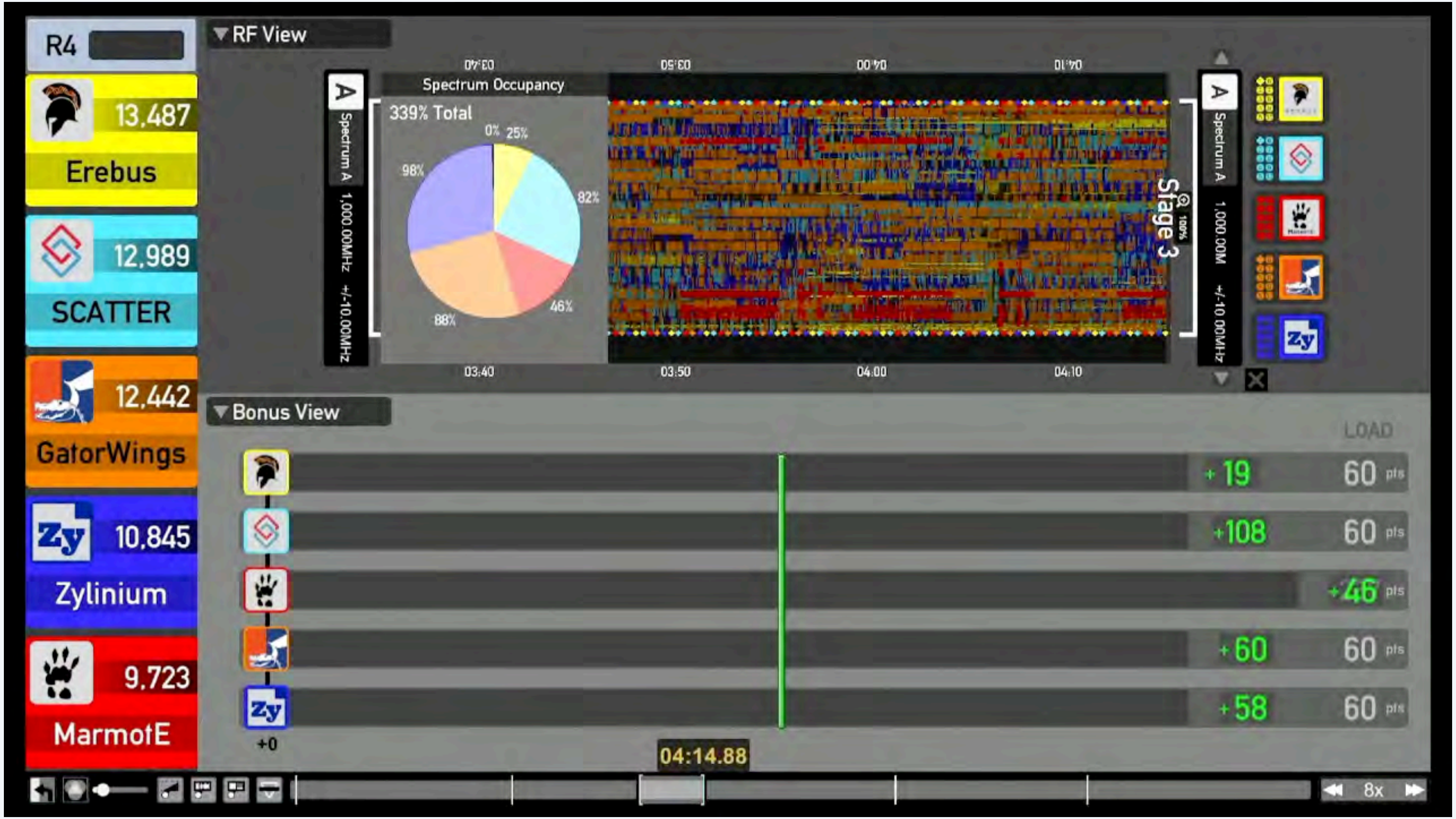
Our Strategy - 3

- Use ML to identify teams
- Determine when we are playing against a set of “Best Teams”: MarmotE, Zylinium , Erebus

- **Championship Mode:** Be ultra-aggressive whenever see all these teams in a match
  - **Use extra channels to allow data to be sent with more robust modulation and allow for more and faster retransmissions**
  - **Jam other teams aggressively whenever we think those teams are potentially outscoring us**

SCE Round 4: Slice of Life

SCE Round 4: Slice of Life



Lessons Learned

- Start from robust, adaptive radio
- Essential to optimize strategy based on objective (scoring rules) —
- Competition changed from cooperative to competitive
  - Tie-breaker scoring for elimination rounds made being aggressive a good strategy
  - Added many features in last 2 weeks to make our radio more aggressive and robust:
    - **Redundant channels**
    - **Jamming**
    - **Championship mode**

Lessons Learned

- No machine-learning "kumbaya" black box that can solve spectrum access problem
- Domain-specific engineering to decompose problem
- Not enough training and validation data for ML
    - Need a less resource-intensive simulation environment to train ML algorithms
- Peer strategies (and probably radios) rapidly changed during last few weeks of SC2
    - ML algorithm (operation after training) couldn't catch up
    - Switched to ES algorithm
    - Probably need more exploration and switching system to cope with rapid updates

Open Research Problems

- How to train multi-agent reinforcement learning agents with large state and action spaces? Need simulation environment and new algorithms?
- How to ensure ML solutions are robust when new situations are encountered?
- How to set up a "scoring" metric that encourages collaboration and responsible behaviors?
- How to make CIL practical?
  - How to protect privacy/security of participants? (Geolocation information, competition-sensitive performance info)
  - How to ensure truthfulness? (Participants may lie about spectrum use, priorities, performance.) How to police spectrum usage and who should do it?
  - How to assign priorities across (or even within) coalition teams?

Acknowledgements

- 2 million thanks to our student team members for their dedication and hard work!
- DARPA SC2 Team for running SC2 and developing Colosseum
- DARPA SC2 Prizes and NSF EAGER Grant 1738065 for supporting our team's efforts
- L3Harris for loaning us two Ettus X310 radios
- Our families for putting up with us during the past 3 years

  "Luckily, we are all still married." – Miklos Maroti, Team MarmotE

Thank You!