

# Adaptive Wireless Networks for Spectrally Efficient Communication<sup>1</sup>

DARPA SC2: Phase 2 Technical Paper  
Team BAM! Wireless

## Contents

<b>1</b>	<b>Summary</b>	<b>4</b>
<b>2</b>	<b>Introduction</b>	<b>4</b>
<b>3</b>	<b>Methods, Assumptions, and Procedures</b>	<b>4</b>
3.1	Physical Layer Protocol . . . . .	5
3.1.1	Frame Header . . . . .	6
3.1.2	Frame Payload . . . . .	6
3.1.3	Adaptive MCS Scheme . . . . .	6
3.1.4	Transmit Power . . . . .	7
3.2	Network Layer and Routing . . . . .	8
3.2.1	Multi-Hop Routing . . . . .	8
3.2.2	Automatic Repeat Request (ARQ) . . . . .	8
3.3	Flow Scheduling . . . . .	8
3.3.1	Static Flow Scheduling for TDD . . . . .	9
3.3.2	Static Flow Count Maximization . . . . .	11
3.3.3	Deficit Round Robin . . . . .	12
3.4	Control Channel . . . . .	12
3.5	Channel Allocation . . . . .	13
3.6	Collaboration . . . . .	14
3.7	Issues and Challenges . . . . .	15
<b>4</b>	<b>Results and Discussion</b>	<b>16</b>
4.1	Overview . . . . .	16
4.2	Spectrum Reuse . . . . .	16
4.3	Passive and DSRC Incumbent Protection . . . . .	16
4.4	Role of Collaboration . . . . .	17
4.5	Payline Success . . . . .	17
4.6	Issues and Challenges . . . . .	17
<b>5</b>	<b>Phase 3 Planned Changes</b>	<b>18</b>
5.1	Machine Learning Framework and Data Generation . . . . .	18
5.1.1	Deep Learning Model . . . . .	19
5.1.2	Dataset Generation . . . . .	19
5.2	Augmenting Current Solutions . . . . .	19
5.2.1	Channel Allocation . . . . .	19
5.2.2	Flow Scheduling . . . . .	20

---

<sup>1</sup>The views, opinions, and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

Approved for Public Release, Distribution Unlimited.

5.2.3	MCS Selection . . . . .	20
5.2.4	Context Understanding . . . . .	21
<b>6</b>	<b>Conclusions</b>	<b>21</b>

## List of Figures

1	BAM! Wireless Network for the DARPA SC2 [1, Figure 1]. . . . .	4
2	High-level diagram showing the different layers and functional components of the BAM! Wireless CIRN [1, Figure 2]. The numbered features in diamonds are: 1- Multi-Hop Routing, 2-Transmit Scheduling, 3-Broadcast of Control Data, 4-High Rate Data Link, 5-Transmit Power Control, 6-Adaptive Modulation and Coding Scheme, 7-Automatic Repeat Request, 8- Program Configuration through Database, 9- Visualization of Performance Measures, 10- Channel Allocation Algorithm. . . . .	5
3	Power spectrum of employed waveform [1, Figure 6]. . . . .	6
4	Block error rates at different SNR levels for all combinations of supported pairs of modulation scheme and code rate [1, Figure 7]. . . . .	6
5	Adaptation of modulation order and code rate due to change in estimated noise variance [1, Figure 8]. . . . .	7
6	The MCS choice has a minimum distance between constellation points that is closest to the diameter of the uncertainty ball created by Gaussian noise. . . . .	9
7	Example of deficit round-robin flow queue scheduling where flow queues $F\#$ with credits ( $\#$ ) containing packets $P\#$ of size ( $\#$ ) are scheduled [1, Figure 3]. . . . .	10
8	An example of a graph of flow nodes $\mathcal{B}_{V,i}$ . Each node contains the indices $i$ where $b_i = 1$ in $\mathcal{B}_{V,i}$ . The effective minimum throughputs are sorted by $\tilde{R}_0 \geq \tilde{R}_1 \geq \tilde{R}_2 \geq \tilde{R}_3$ and delays $\alpha_3 L_3 \leq \alpha_2 L_2 \leq \alpha_1 L_1 \leq \alpha_0 L_0$ . A red line connects a parent to its child node with the greatest $\tilde{R}_i$ removed by setting $b_i = 0$ in the child and, likewise, a blue line indicates the least $\alpha_i L_i$ removed. Dashed nodes and lines are not visited. . . . .	12
9	Robust synchronization method developed for the low signal-to-interference ratio scenario of the DARPA Spectrum Challenge leveraged for the BAM! wireless robust control link [1, Figure 4]. . . . .	13
10	Control transmissions observed in SRN power spectrum [1, Figure 5]. . . . .	14
11	Occupied spectrum in an Alleys of Austin scenario [1, Figure 9]. . . . .	15
12	Captures from our GUI showing our spatial frequency reuse. . . . .	17
13	Captures from our GUI showing our protection of the passive incumbent. . . . .	18
14	Captures from our GUI showing our collaboration to maximize the minimum score across the ensemble. . . . .	19

## 1 Summary

A description of the key technical aspects of the design of the BAM! Wireless Collaborative Intelligent Radio Network (CIRN) that was developed in Phase 2 of the DARPA SC2 Challenge is provided in this paper. The presented design adapts intelligently to the existing Colosseum scenario as well as the behavior of peer teams, in a robust manner that takes into account a wide range of error or unexpected event possibilities. As illustrated in the sequel, reaching this solution required innovating lightweight intelligent capabilities on top of flexible and robust physical and network layers.

## 2 Introduction

We present in this paper important aspects of the technical design of the BAM! Wireless Collaborative Intelligent Radio Network (CIRN) that participated in the second preliminary event of DARPA SC2. The CIRN consists of multiple Standard Radio Nodes (SRN) and one Gateway node, as depicted in Figure 1. All nodes interact directly with the Colosseum emulator, while only the gateway can send and receive collaboration messages according to the defined CIRN Interaction Language (CIL).

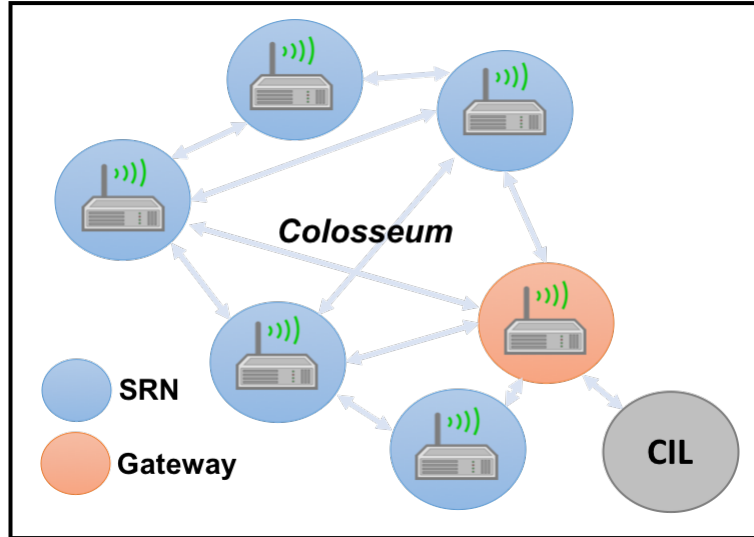


Figure 1: BAM! Wireless Network for the DARPA SC2 [1, Figure 1].

Following the Phase 2 scoring procedure, the approach behind the BAM! Wireless CIRN design is centrally focused on maximizing the minimum number of achieved mandates by a team in any given match (ensemble minimum), by regularly monitoring the reported performance and use of resources of peer teams through the received collaboration messages. The CIRN design specifically relies on lightweight intelligent algorithms that are robust to changing environmental conditions. The final outcome is a robust intelligent solution that jointly optimizes the decisions for the modulation and coding scheme, flow scheduling, as well as channel allocation to maximize the ensemble performance in the wide range of Colosseum scenarios. We next describe in detail the various technical components of the CIRN.

## 3 Methods, Assumptions, and Procedures

We describe in this section the different components and tasks involved in the CIRN, and where each task is taking place across the layered architecture. We provide a high-level description in Figure 2.

Note that the *Network Controller* function pertains only to the Gateway node; all other components and features are common to all SRN nodes as well as the Gateway. We first start by describing the physical layer protocol below.

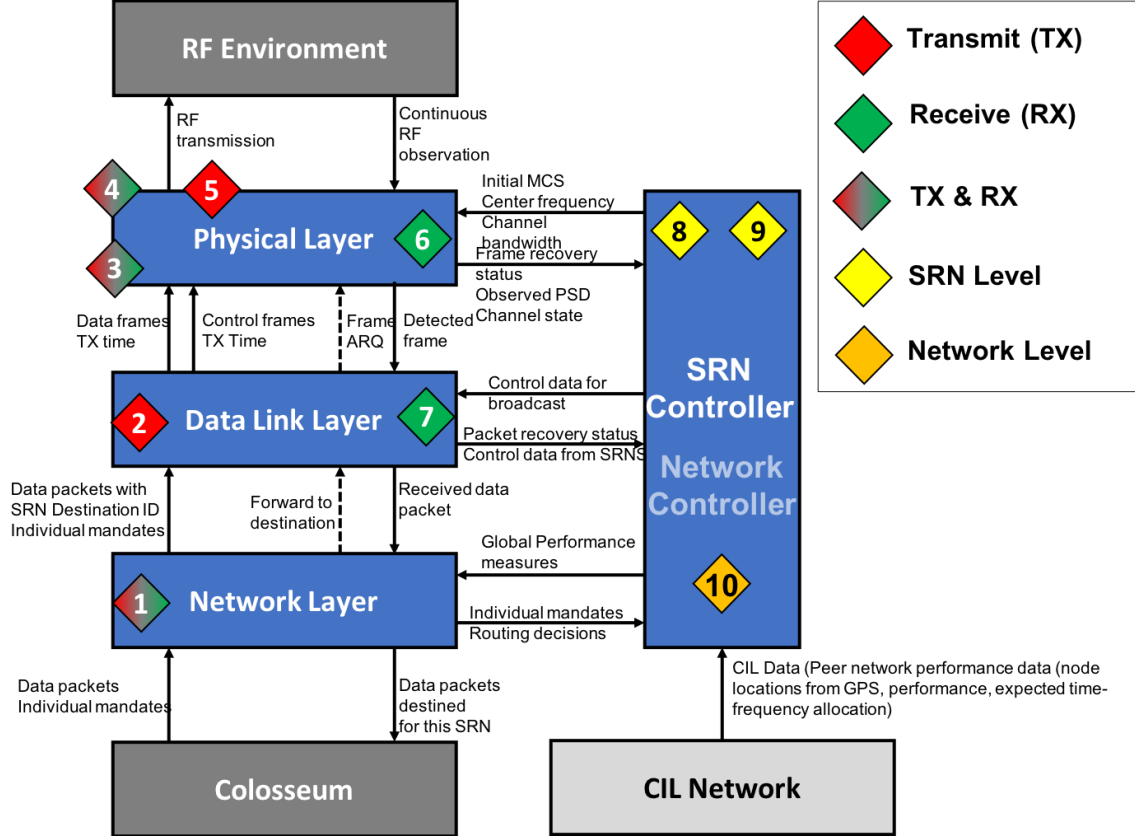


Figure 2: High-level diagram showing the different layers and functional components of the BAM! Wireless CIRN [1, Figure 2]. The numbered features in diamonds are: 1- Multi-Hop Routing, 2-Transmit Scheduling, 3-Broadcast of Control Data, 4-High Rate Data Link, 5-Transmit Power Control, 6-Adaptive Modulation and Coding Scheme, 7-Automatic Repeat Request, 8- Program Configuration through Database, 9- Visualization of Performance Measures, 10- Channel Allocation Algorithm.

### 3.1 Physical Layer Protocol

As described in [1], we use a DFT-s-OFDM waveform with different choices for the bandwidth, center frequency, and power spectrum (see Figure 3). Each channel is defined by a choice of a center frequency and bandwidth pair, and each SRN is assigned a channel for transmission at any given point in time. Further, each SRN can listen to all other nodes in the network through multiple *receive chains*. Each waveform occupies 108 subcarriers for data symbols, 12 subcarriers for pilot symbols, and 8 subcarriers that are used for estimating the noise variance. An algorithm similar to Schmidl & Cox time synchronization is used by all receivers. The used modulation types are QPSK, 16-QAM, 32-QAM, and 64-QAM. The error correction codes are selected from the IEEE 802.11 QC-LDPC suite of codes with rates  $1/2$ ,  $2/3$ ,  $3/4$ , and  $5/6$ . In Figure 4, we plot the simulated block error rates for different choices of the modulation scheme and code rate at a wide range of SNR values.

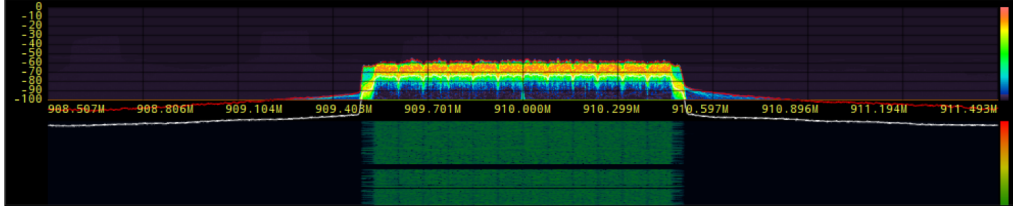


Figure 3: Power spectrum of employed waveform [1, Figure 6].

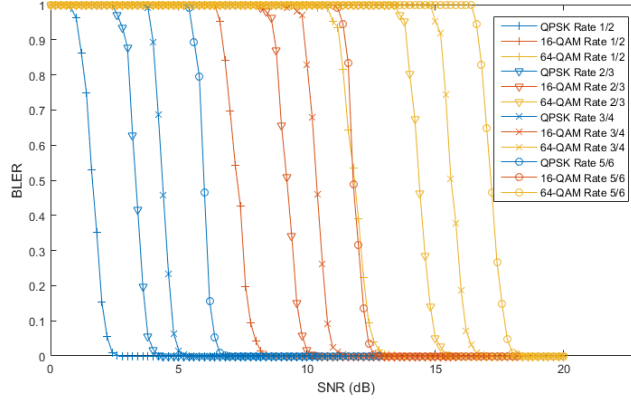


Figure 4: Block error rates at different SNR levels for all combinations of supported pairs of modulation scheme and code rate [1, Figure 7].

### 3.1.1 Frame Header

We use QPSK modulation and a rate 1/2 code for frame headers. A frame header includes the source and destination identifiers, the modulation and coding scheme index, and the number of blocks carried in the frame payload. Frame headers also include latency critical control and feedback information; currently, ARQ feedback is included. To further minimize the probability of not correctly identifying the frame header, an extra protection is provided using a 32-bit cyclic redundancy code. Finally, the header also contains a frame identification number, which is used for data logging and analysis.

### 3.1.2 Frame Payload

A pair of modulation scheme and coding rate is selected for the frame payload based on the current environment, as described below. The blocks within the frame payload are mapped to subcarriers of a sequence of OFDM symbols. The content of a block originate from segments, which can be either IPv4 segments carrying Colosseum data, segments carrying automatic repeat request information, or control segments.

### 3.1.3 Adaptive MCS Scheme

The employed adaptive MCS scheme is based on finding the smallest Euclidean distance between the boundaries of circles centered around constellation symbols. Each of these circles has a radius that is proportional to the ratio of estimated *noise+interference* standard deviation and asymptotic code gain. The following factors are further considered:

1. A lower bound on throughput is respected.

2. An upper bound on block error rate is respected.

For the second constraint above, a closed form approximation is assumed for computing the block error rate given the minimum distance between constellation points and noise standard deviation.

Note that the noise variance estimate relies on a sliding window median filter to remove outliers. In Figure 5, we show an example for the illustrated adaptive MCS scheme on a Colosseum payline scenario. It is important to highlight that the link rate reduces as the measured standard deviation of the additive noise increases. See Algorithm 1 for a pseudo code of the adaptive MCS algorithm.

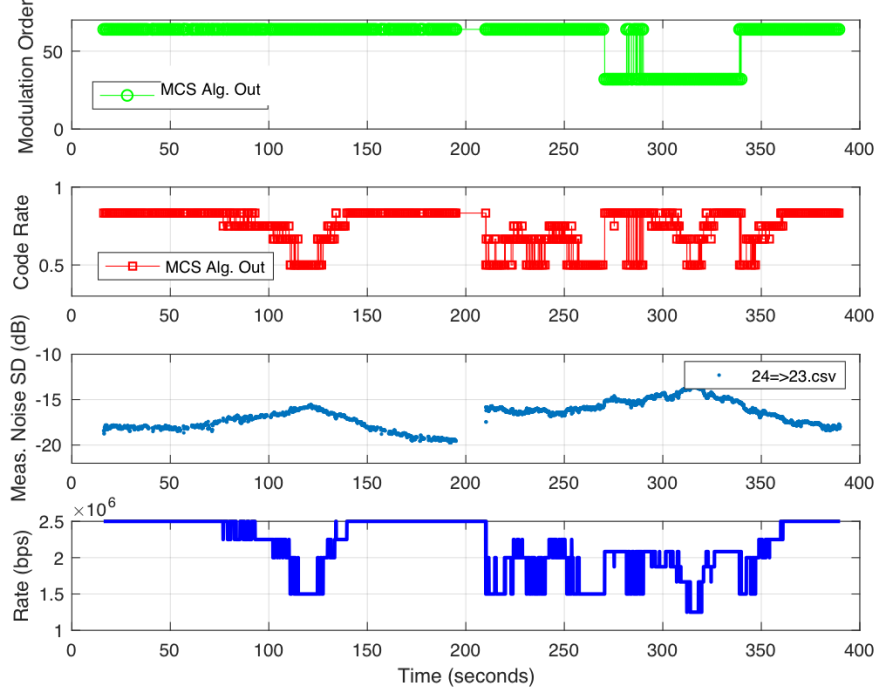


Figure 5: Adaptation of modulation order and code rate due to change in estimated noise variance [1, Figure 8].

### 3.1.4 Transmit Power

Our system uses one of the antennas for the data channel and the other for the control channel. Each transmission can only use half of the total power available to the radio. By default, we always transmit at maximum power. We only reduce our transmit power when we need to protect incumbents. This is done as follows.

A node-based transmit power control is employed for passive incumbent protection. Each incumbent regularly broadcasts a CIL message that contains its current channel information, measured total receive power, and a receive power threshold. When the measured power violates the threshold, the CIRN nodes should collectively lower transmit power levels, such that the violation ceases to exist at the incumbent. On the other hand, when the threshold is not violated, nodes are allowed to increase their transmit power levels while collectively respecting the incumbent constraint. Here, collaborative CIL messages are crucial to determine the total number of nodes - over all participating CIRNs - that are causing interference at the incumbent receiver.

Assuming non-coherent power combining at the receiver of the incumbent, the transmit power level for each SRN that causes interference at the incumbent is lowered, by an amount proportional

---

**Algorithm 1** Pseudo Code for the Adaptive MCS Algorithm

---

- 1: Estimate noise variance  $\sigma$  using *spectrum holes*.
  - 2: Noise variance estimate is median filtered using the past one second measurements to obtain the new estimate  $\tilde{\sigma}$ .
  - 3: Using the filtered noise estimate, block error rate is estimated for each considered choice of modulation scheme and code rate.
  - 4: Any MCS choice leading to a block error rate above an empirically determined upper bound is not considered.
  - 5: Among the remaining choices, choose the modulation scheme  $m$  and coding rate  $r$  that minimize  $B(m, r) = \left| \alpha d(m) - \frac{2\tilde{\sigma}}{\sqrt{G_c(r)}} \right|$ , where  $\alpha$  is an empirically determined parameter,  $d(m)$  is the minimum distance between a pair of constellation points, and  $G_c(r)$  is the error control code gain. See Figure 6 for an illustration of this step.
  - 6: Via control messaging, the receiver informs the transmitter of the new MCS choice.
- 

to the violated distance from the threshold, and assuming that other networks will follow a similar procedure to lower the transmit power levels of nodes interfering at the incumbent receiver.

In PE2 we lowered our transmit power for up to 10dB, but did not stop transmissions entirely. In retrospect, this proved to be a mistake that we plan to fix in Phase 3.

### 3.2 Network Layer and Routing

Here, we describe the multi-hop routing and automatic repeat request capabilities supported by the network layer.

#### 3.2.1 Multi-Hop Routing

We detect the absence of a link between a pair of nodes through the absence of fresh control information. A binary routing table is then constructed after each node shares its local connectivity information. Dijkstra's algorithm is then applied to the routing table, in order to find the shortest path between the source and destination of a flow.

#### 3.2.2 Automatic Repeat Request (ARQ)

Reliable flow transport is provided by a basic automatic repeat request (ARQ) mechanism based on sequence acknowledgements and a sliding-time window. First, we evaluate the reliability requirements of each flow, and determine whether automatic repeat request is needed. ARQ is enabled for all file flows for the CIRN that participated in the second preliminary event. ARQ for stream flows did not perform to standards and was disabled for the preliminary event 2.

### 3.3 Flow Scheduling

As different flows have different rate and latency requirements, we employ an adaptive scheduling algorithm that assigns *time quanta*, and applies a deficit round-robin scheduling algorithm. Time quanta are assigned, such that a flow is scheduled if there is a good chance that it meets requirements. The applied deficit round-robin flow scheduling mechanism ensures fairness of the assigned time quanta. Moreover, prioritization and anti-starvation is incorporated into the round-robin mechanism (thus allowing deviations from traditional and incorporation of ARQ for flow queuing (see Figure 7 for an illustration)).



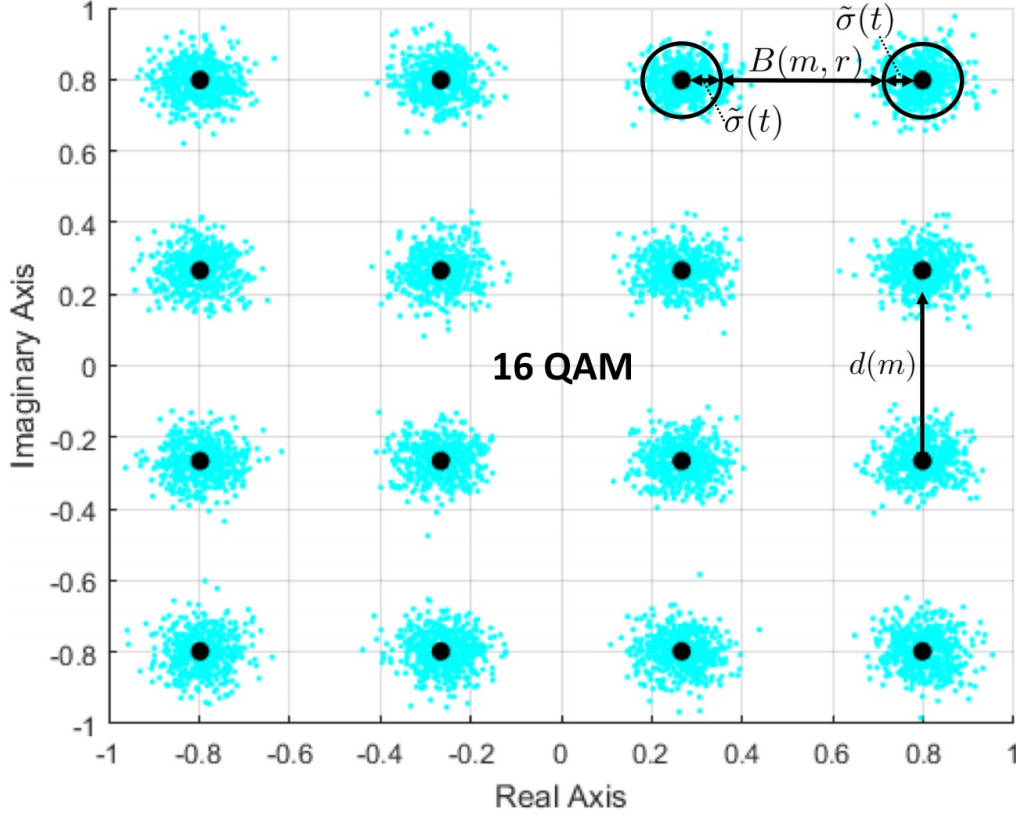


Figure 6: The MCS choice has a minimum distance between constellation points that is closest to the diameter of the uncertainty ball created by Gaussian noise.

In what follows, we shed light on more details of our flow scheduling algorithm. Note that the scoring function used in Phase 2 seeks to maximize the minimum network score. Further, network scores can be summarized by the number of network flows that a network transports within a set of constraints. One set of constraints are the *gates* for spectrum usage, which are penalty terms and are considered outside the scope of local flow scheduling. The set of constraints on flows considered in this section concern the quality of service (QoS) metrics of delay and throughput.

The following outlines the basic algorithms for locally optimizing the schedule of flows at a single node in the BAM! Wireless CIRN.

### 3.3.1 Static Flow Scheduling for TDD

We show here how to determine if a given set of flows may be scheduled for transmission on a single time-division duplexed (TDD) link at a node. A flow is scheduled only if its QoS constraints can be satisfied for the affected measurement windows to follow. We assume that many parameters are fixed or specified to simplify the solution. Other components of flow scheduling build on the following static flow scheduling algorithm. Outputs of the algorithm are a yes or no if a given combination of flows can be scheduled (the decision problem) and if true, sufficient time allocations (termed time quanta) for each flow to satisfy its QoS constraints.

At a network node, a flow has a queue length  $q_i(t)$  at time  $t$ , a maximum packet delay  $L_i$  (seconds), a minimum throughput  $R_i$  (bits per second), an expected packet size  $M_i$  (bits), and a minimum number of packets per transmit frame  $J_i$ . Queue length  $q_i$  excludes packets that will be dropped for being too late upon network egress. At any point in time at a node, from the set of

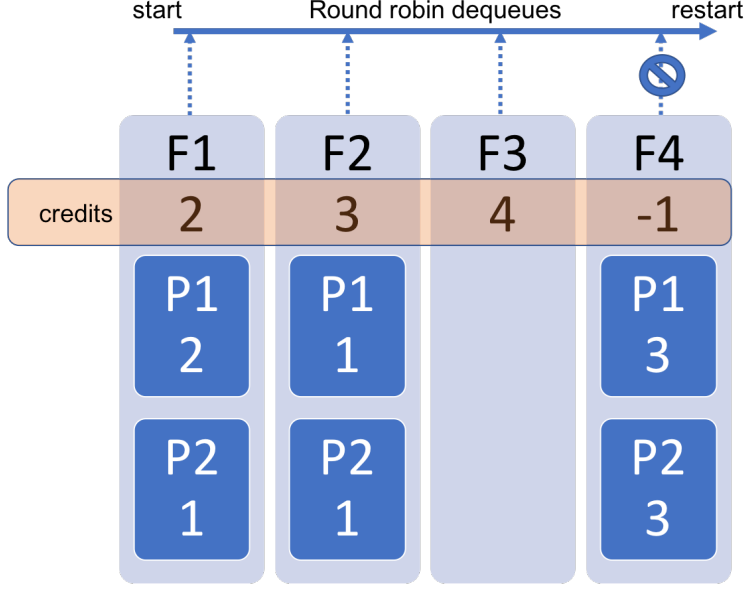


Figure 7: Example of deficit round-robin flow queue scheduling where flow queues  $F\#$  with credits  $(\#)$  containing packets  $P\#$  of size  $(\#)$  are scheduled [1, Figure 3].

all flows  $\mathcal{F}$ , the subset of active flows  $\mathcal{A} \subset \mathcal{F}$  is the set of flows with a non-zero queue length  $q_i$ , i.e.,  $\mathcal{A} = \{f_i \in \mathcal{F} : q_i > 0\}$ .

Several properties of the TDD link affect the scheduling of flows. Packets of a flow (ideally  $J_i$  at a time) are framed for physical layer transmission. Framing incurs an overhead of  $G_i$  seconds per frame and  $S$  bits per packet. The transmission rate of the link is  $x_i(t)$  (bits per second) with an expected codeword error rate of  $e_i(t)$  at time  $t$ . Thus, an approximate value for the expected goodput (throughput less errors) is  $y_i(t) = x_i(t)[1 - e_i(t)]$  and ARQ goodput (goodput less retransmissions) is  $z_i(t) = x_i(t)[1 - e_i(t)]$ .

For some subset of active flows  $\mathcal{C} \subset \mathcal{A}$ , the scheduler assigns time quanta  $a_i$  for the flows  $f_i \in \mathcal{C}$  that minimize the total time it takes to serve the flows in  $\mathcal{C}$  while ensuring that packets in the flows are all serviced according to their QoS constraints. The following linear program expresses the objective and constraints.

$$\arg \min_{a_i} \sum_i a_i \quad (1a)$$

$$\text{subject to } a_i \geq \gamma_i, \quad f_i \in \mathcal{C} \quad (1b)$$

$$\sum_i a_i \leq \min_{j \in \mathcal{L}} \alpha_j L_j \quad (1c)$$

$$a_j \geq \tilde{R}_j \sum_i a_i, \quad j \in \mathcal{R} \quad (1d)$$

The overhead of a frame of packets for a flow is

$$\gamma_i = G_i + J_i \frac{S + M_i}{x_i}, \quad (2)$$

and the minimum effective rate after accounting for overhead is

$$\tilde{R}_i = \frac{R_i x_i \gamma_i}{y_i J_i M_i}. \quad (3)$$

We assume a flow's packets cannot be divided into less than  $\gamma_i$  bits per framed transmission per constraint (1b). Furthermore, the minimum of flows' maximum delays ( $\mathcal{L} = \{i : f_i \in \mathcal{C}, L_i < \infty\}$  is the set of indices of flows with maximum delays) bounds the total time allotted per constraint (1c). Scaling delays by  $\alpha_j$  allows expressing tighter delay constraints due to any additional queueing, transmission, and processing time. Finally, (1d) expresses that the time quantum  $a_j$  should be sufficiently long to attain the minimum throughput  $R_j$  for any flow with a minimum egress throughput constraint ( $\mathcal{R} = \{i : f_i \in \mathcal{C}, R_i > 0\}$  is the set of indices for such flows with throughput constraints).

The many parameters that affect the solution are assumed known and fixed so we consider the optimizers  $\{a_i^*\}$  of (1) as a *static* schedule for the flows on a TDD link. If a solution  $\{a_i^*\}$  exists, then the set of flows  $\mathcal{C}$  can be serviced in a TDD fashion with  $a_i^*$  amount of time allotted to flow  $f_i \in \mathcal{C}$ .

Our code solves (1) by performing a binary search on the range between the upper and lower bounds established on  $\sum_i a_i$  by (1b,1c,1d).

### 3.3.2 Static Flow Count Maximization

With a solution to the question of whether a set of flows can be scheduled for transmission while satisfying their QoS constraints, we now examine how to find the largest sized set of flows. By finding the largest sized set of flows that can be scheduled, we implicitly also find the highest value because all flows have equal value once they reach their hold times according to Phase 2 scoring. The problem is combinatorial in nature and exhaustive search quickly becomes infeasible.

If, instead of (1a), we consider maximizing the number of serviced flows, we can objectify and optimize for the number thereof. First, observe that the objective function (1a) was chosen simply to minimize the scheduling time for all chosen flows to benefit from certain properties of our round robin scheduling implementation. Consider the candidate set of flows  $\mathcal{C} = \mathcal{A}$  of all  $|\mathcal{C}| = N$  active flows and substitute maximizing the number of scheduled flows for (1a). Let  $b_i = 1$  if flow  $f_i$  is scheduled. Then, statically optimizing for the flow count may be expressed by

$$\max_{b_i, a_i} \sum_i b_i \quad (4a)$$

$$\text{subject to } b_i a_i \geq b_i \gamma_i, \quad f_i \in \mathcal{C} \quad (4b)$$

$$\sum_i b_i a_i \leq \min_{j \in \mathcal{L}, b_j=1} \alpha_j L_j \quad (4c)$$

$$b_j a_j \geq b_j \tilde{R}_j \sum_i b_i a_i, \quad j \in \mathcal{R} \quad (4d)$$

$$b_i \in \{0, 1\}, \quad f_i \in \mathcal{C}. \quad (4e)$$

The constraints now include the  $b_i$  decision terms where appropriate to activate constraints. An exhaustive search over all combinations of flows quickly grows infeasible as the total number of possible combinations of  $N$  flows is  $\sum_{k=1}^N \binom{N}{k} = 2^N - 1$ .

To approach the problem, we construct a graph (Figure 8) where nodes represent unique sets defined by

$$\mathcal{B}_{V,i} = \{b_j = \{0, 1\} : j = 1, \dots, N; \sum_j b_j = V\}, \quad (5)$$

where  $V$  flows are scheduled and  $i \in \{1, \dots, \binom{N}{V}\}$  indexes the  $\binom{N}{V}$  flow combinations. Node  $\mathcal{B}_{V-1,i}$  is a child of  $\mathcal{B}_{V,i}$  if it has all the same scheduled flows as its parent except one. The node representing  $\mathcal{B}_{N,1}$  has no parents and will be the "root" node.

We search over the graph of nodes  $\mathcal{B}_{V,i}$  by starting at the root node and descending down in turn to the lower  $V$  levels. Since each node at a constant  $V$  level is equal in value for scoring, any would

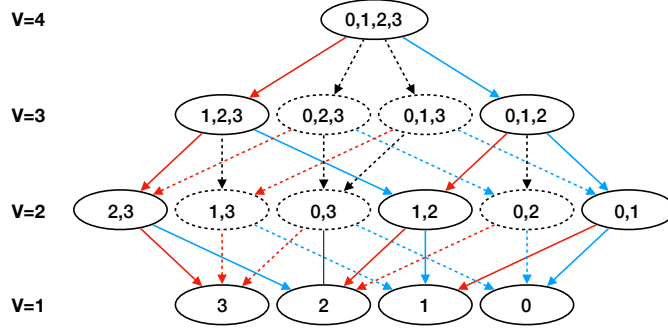


Figure 8: An example of a graph of flow nodes  $\mathcal{B}_{V,i}$ . Each node contains the indices  $i$  where  $b_i = 1$  in  $\mathcal{B}_{V,i}$ . The effective minimum throughputs are sorted by  $\tilde{R}_0 \geq \tilde{R}_1 \geq \tilde{R}_2 \geq \tilde{R}_3$  and delays  $\alpha_3 L_3 \leq \alpha_2 L_2 \leq \alpha_1 L_1 \leq \alpha_0 L_0$ . A red line connects a parent to its child node with the greatest  $\tilde{R}_i$  removed by setting  $b_i = 0$  in the child and, likewise, a blue line indicates the least  $\alpha_i L_i$  removed. Dashed nodes and lines are not visited.

be a suitable choice for scheduling. One further optimization we make is to reuse computation from parent nodes when determining the feasibility of child nodes. Furthermore, certain child nodes need never be visited (based on their constraints and computation from their parent nodes) and thus large parts of the graph can be eliminated from consideration.

### 3.3.3 Deficit Round Robin

A solution to (4) consists of a set of scheduled flows  $\mathcal{B}_{V,i}$  and their time quanta  $\{a_i\}$ . The time quanta are used in the deficit round robin scheduling scheme of [2]. Although rather involved in its implementation, the basic principle of deficit round robin is well established and even included in the Linux kernel. The primary advantage we attain from employing deficit round robin is a guarantee on satisfying QoS constraints given our solution to (4).

## 3.4 Control Channel

We employ two types of control messages:

1. Short control messages are transmitted with a modulation type of 8-FSK and a bandwidth of 480 KHZ. A time-hopping synchronization scheme is specially used for these control messages (see Figure 9). Also, a Reed-Solomon error control coding is used. The center frequency used for short control messages alternates every second between the upper and lower band edges<sup>2</sup>.

We use a variant of the Slotted ALOHA multiple access mechanism without collision detection and backoff, for short control messages. Every slot period is 60 ms. Nodes transmit short control messages at slot boundaries with a uniform probability whose value is inversely proportional to the number of nodes in the CIRN. The short control message carries information about the number of nodes in the network, the allocated channels, and a timestamp. Nodes broadcast modified states, upon receiving freshly updated short control messages. The advantage of having short control messages is to use it for initial node discovery, and as a reliable backup option when the interference level is high.

2. Long control messages are carried within the DFT-s-OFDM data links. In addition to the basic information carried by short control messages, these long control messages carry information

<sup>2</sup>This special allocation for short control messages caused problems during the second preliminary event when there was a badly performing peer team that consistently requested overlapping channels; we are investigating how to fix this issue for Phase 3.

about traffic statistics and measured performance of individual mandates.

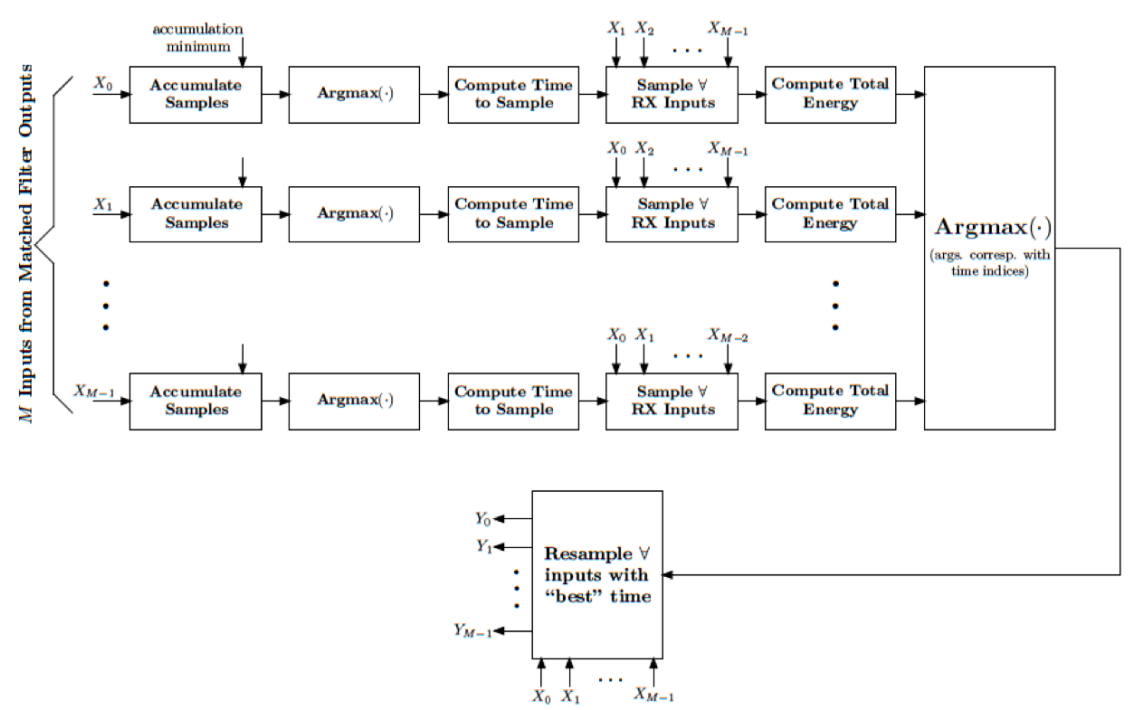


Figure 9: Robust synchronization method developed for the low signal-to-interference ratio scenario of the DARPA Spectrum Challenge leveraged for the BAM! wireless robust control link [1, Figure 4].

Figure 10 shows the spectrogram that one of our nodes measured during a qualification scenario. Observe that short control messages occupy the narrow bands at the lower and upper edges.

### 3.5 Channel Allocation

The gateway node first estimates the channel gain between each of our nodes and all others, assumed to be frequency-independent, symmetric, and slowly varying. These gains are initialized using the GPS locations reported through the CIL and assuming free space propagation with loss exponent of 3. These values are then refined adaptively based on our own PSD measurements and CIL messages from other nodes' specifying their transmit power and frequency. The channel gain estimates and the spectrum voxel collaboration messages from other teams are hence used to construct an interference map for each frequency.

The interference spectrum is different for each of our nodes. The gateway node goes through all the frequency bands and determines the value that each band would have for each node. This value function takes into account the interference maps as well as the flows that each node needs to deliver. Then, it assigns each node a channel (center frequency and bandwidth) to transmit on. This choice is made through a combination of a greedy algorithm that attempts to maximize the sum of the values and a *considerate* algorithm that reallocates channels to help improve the ensemble minimum. The channel allocation mechanism responds to environmental updates and changes in offered mandates. If a node needs to transmit more traffic to a specific receiver, or the flows have more stringent requirements, the gateway will take that into account when picking the center frequency.

It is important to highlight that an obstacle we faced towards making the channel allocation algorithm more biased towards a collaborative behavior was that the information that we were receiving from peer nodes (specifically, the GPS locations and SpectrumUsage) was frequently inaccurate.

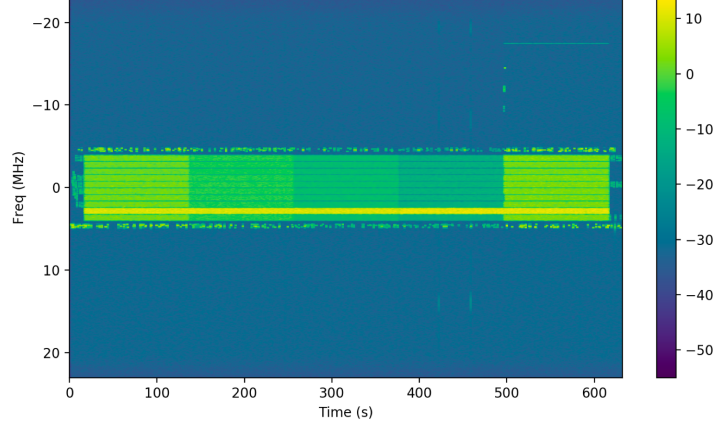


Figure 10: Control transmissions observed in SRN power spectrum [1, Figure 5].

In summary, each of our nodes will receive on multiple channels, but transmit on a single one (apart from the control channel). The bandwidth required for each node is chosen based on the offered load. We plan to incorporate the new scoring mechanism of Phase 3 to bias the channel allocation mechanism towards a more greedy behavior in the stage where extra individual mandates contribute to score without regard to the ensemble. See Algorithm 2 for a pseudo code of the channel allocation algorithm.

---

**Algorithm 2** Pseudo Code for Channel Allocation Algorithm - Routine call every one second

---

- 1: **if** We are the worst performing team for 5 consecutive routine calls. **then**
  - 2:     Call Channel Re-allocation Algorithm.
  - 3: **else if** We are two mandates above the worst performing team for 5 consecutive routine calls.
  - 4:     Reduce bandwidth allocated for widest channel.
  - 5: **end if**
- 

We show in Figure 11 an example behavior of the channel allocation mechanism in an Alleys of Austin scenario.

### 3.6 Collaboration

Although our PE2 system was not able to take full advantage of the information received through the CIL, we attempted to report as much information as we could about our state and plans to other teams. Unfortunately, our gateway did not always have access to all the details, so we had to make several approximations. For example, the number of individual mandates (IMs) that we were achieving and the duty cycle with which we were using a specific band were often inaccurate because they were computed in a distributed manner across all the nodes.

Our system did rely heavily on three CIL messages: `spectrum_usage`, `location_update`, and `detailed_performance`. Our decision engine kept track of the latest reports from each network and used them to gauge our performance and allocate our channels. Specifically, we attempted to maximize the minimum score across the ensemble by reducing the bandwidth of our channels when we were doing well (at least two IMs above the minimum) and hoarding more bandwidth or relocating our channels entirely when we had been the worst performing for a certain period (5 sec. in PE2).

---

**Algorithm 3** Pseudo Code for Channel Re-allocation Algorithm

---

**Input:** Vector of Bandwidths assigned to receivers.

$IP(j, f)$ : Estimated interference power at node  $j$  and frequency  $f$ .

$G(k, j) = G(j, k)$ : Channel gain between nodes  $k$  and  $j$ .

$P(k)$ : Transmit power at node  $k$ .

$N(j)$ : The set of neighbors (with significant interference) of receiver at node  $j$ .

$d(k, j)$ : Distance between nodes  $k$  and  $j$ .

$B(i)$ : Bandwidth allocated for transmitter at node  $i$ .

$SNR(i, j, f_c)$ : Estimated Average SNR between transmitter at node  $i$  and receiver at node  $j$  if center frequency  $f_c$  is selected.

$V(i, f_c)$ : Estimated Value of selecting center frequency  $f_c$  for the transmitter at node  $i$ .

$w(i, j)$ : weight proportional to traffic from node  $i$  to node  $j$ .

$BAM$ : Set of indices for nodes in our network.

- 1: For all  $j \in BAM$ , calculate  $IP(j, f) = \sum_{k \in N(j)} \frac{G(k, j)P(k)}{d(k, j)^3}$ .
  - 2: For all  $i, j \in BAM$  and center frequency  $f_c$ , calculate  $SNR(i, j, f_c) = \frac{1}{B(i)} \sum_{f=f_c-\frac{B(i)}{2}}^{f=f_c+\frac{B(i)}{2}} \frac{G(i, j)}{IP(j, f)}$ .
  - 3: For all  $i \in BAM$  and every possible center frequency  $f_c$ , calculate  $V(i, f_c) = \sum_{j \neq i, j \in BAM} w(i, j)SNR(i, j, f_c)$ .
  - 4: Seek allocation  $i \rightarrow f_c(i)$  so as to maximize  $\sum_{i \in BAM} V(i, f_c(i))$  without channel overlap. This is done through a greedy depth first search in decreasing order of required bandwidth.
- 

### 3.7 Issues and Challenges

During free play matches, we observed that not all nodes reported accurate GPS locations. Initially, we thought that this could be due to bugs in our peers' software, but then we realized that it is probably intended to model realistic failures in GPS reception. We therefore refined our system to detect meaningless GPS coordinates (e.g. all zeros) and replace them with a better estimate based on previously reported coordinates and those of other nodes in the same network.

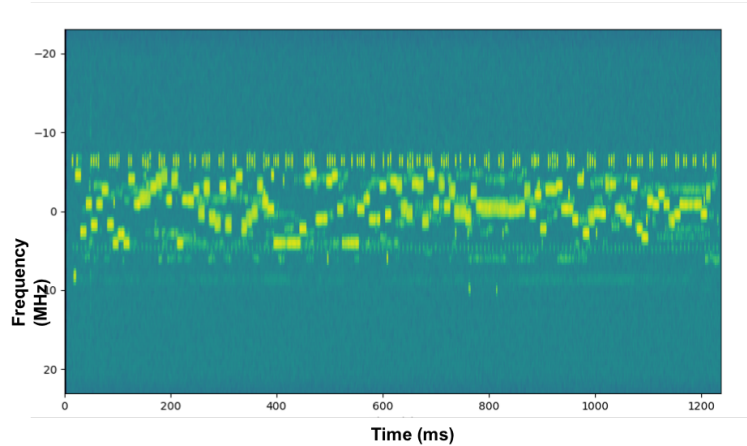


Figure 11: Occupied spectrum in an Alleys of Austin scenario [1, Figure 9].

## 4 Results and Discussion

We discuss our results in PE2, as well as the Payline Redux event, and highlight key aspects that will serve as guidelines for future improvements.

### 4.1 Overview

We have the following general observation from the results of our participation in PE 2 and the Payline Redux events:

1. We observe that we did particularly well - in terms of scoring performance - in Slice of Life scenarios. We are working on further analyzing the logs to understand the reason, but it is likely because of the collaborative behavior that is designed to maximize the ensemble minimum and report accurate spectrum usage information.
2. The *surprise* jammer did not cause severe damage to our CIRN performance.
3. We plan to improve our incumbent protection algorithm, as our CIRN participated in multiple matches where the incumbent threshold was violated for long periods of time.
4. We plan to further improve our adaptive MCS algorithm, as we believe that it played a key role in not passing the payline test during PE 2. Tentative improvements led to our payline redux success, but further improvements are needed in Phase 3.
5. We suspect that having the control channels fixed at the band edges negatively affected peer performance (as discussed in detail in Section 3.4). We plan to improve this by adding an intelligent mechanism that backs up control updates.
6. We plan to further add intelligent capabilities to support channel allocation, in a manner that adjusts to the current scenario and peer behavior.

### 4.2 Spectrum Reuse

Each node is assigned a center frequency and bandwidth as described in Section 3.5. This is done based on the individual interference map from that node to all others in our network, allowing for spatial frequency reuse. Figure 12 illustrates this behavior. However, our MAC layer does not have the capability of preventing nodes from transmitting during specific time intervals, so we were not able to take advantage of time-sharing opportunities.

### 4.3 Passive and DSRC Incumbent Protection

We reacted in the same way for both passive and DSRC incumbents: by progressively reducing our transmit power with each violation report. Figure 13 illustrates this behavior. We assumed that other teams would do the same, but might be slower than us. Therefore, we decided to stop reducing our transmit power once it had dropped 10 dB below our default transmit power (which is not the maximum allowable by the radios, since we are only using one antenna). However, we did not test this hypothesis sufficiently during free play.

From the results of the competition, it seems that our peers were not reducing their power as we were expecting, and that the tolerance threshold after a violation was lower than we expected. Therefore, our performance in the passive incumbent protection scenario was not as good as we expected.



## Payline scenario

At  $t = 237$  competitor reports planned usage of band X for the next 10 sec...

...for transmission between two nodes in a side cluster

Since our node 112 cannot hear those two nodes, it is assigned that same channel. One of our nodes (gateway) might suffer interference, but the other two receivers for 112 will not.

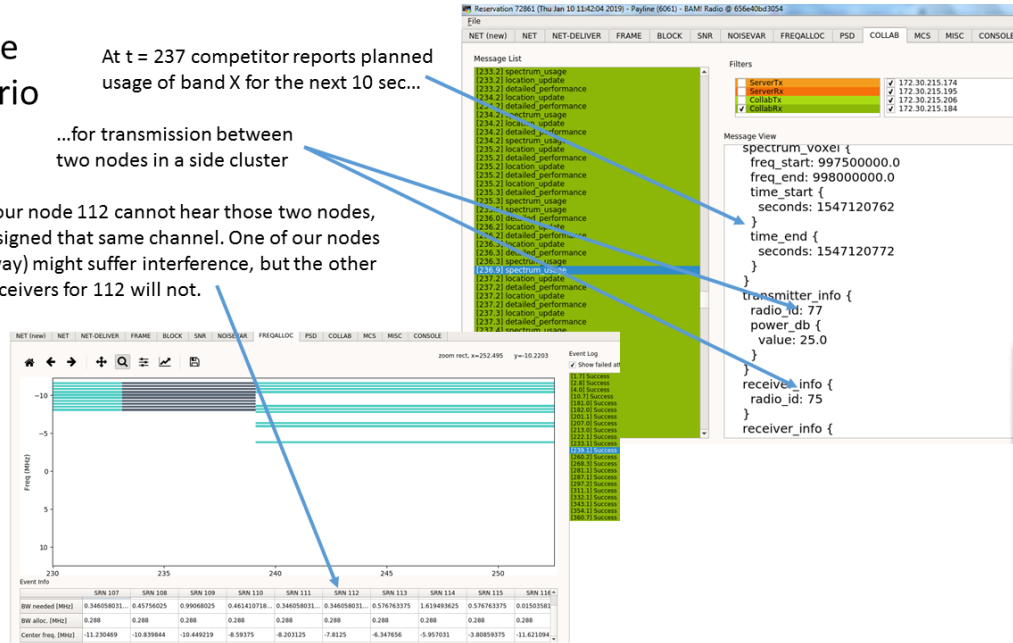


Figure 12: Captures from our GUI showing our spatial frequency reuse.

## 4.4 Role of Collaboration

The different roles that collaboration plays in our system have been described in most of the above sections. However, our ultimate goal in collaborative interactions was to maximize the minimum score across the ensemble at every point in time. Figure 14 illustrates one of the ways in which we used the CIL to achieve that goal through channel reallocation.

## 4.5 Payline Success

We did not succeed in passing the Payline test in PE2, but were the highest ranked team that did not do so, and then successfully passed the test in the Payline Redux event. We believe that the main reasons for our initial failure was the fact that we were being excessively conservative in the adaptation of our channel bandwidths and our MCS (Modulation and Coding Scheme) was not accurately accounting for interference.

Our goal, regardless of the scenario and environment, was to maximize the minimum score across the ensemble. Whenever one of our peers reported a smaller number of IMs achieved than us, we narrowed our channels, trying to give the struggling peer the opportunity to increase their resources. This was usually the case at the beginning of each payline phase, there was always a struggling peer that we needed to help. Eventually, our number of IMs dropped, and then we widened our channels, but it was too late. Most of the bandwidth had already been claimed and the interference was too much for us to handle.

In the Payline Redux event, we changed our approach. We set a lower bound for the width of our channels. Even if others were doing badly, we would not make our channels narrower than 288 KHz. Additionally, we improved our MCS adaptation to better deal with interference.

## 4.6 Issues and Challenges

We did not pass the payline in the PE2, but did pass it in the second opportunity. We believe that our not passing the payline was in part due to the lack of opportunities to run 50 node matches in

# Incumbent protect

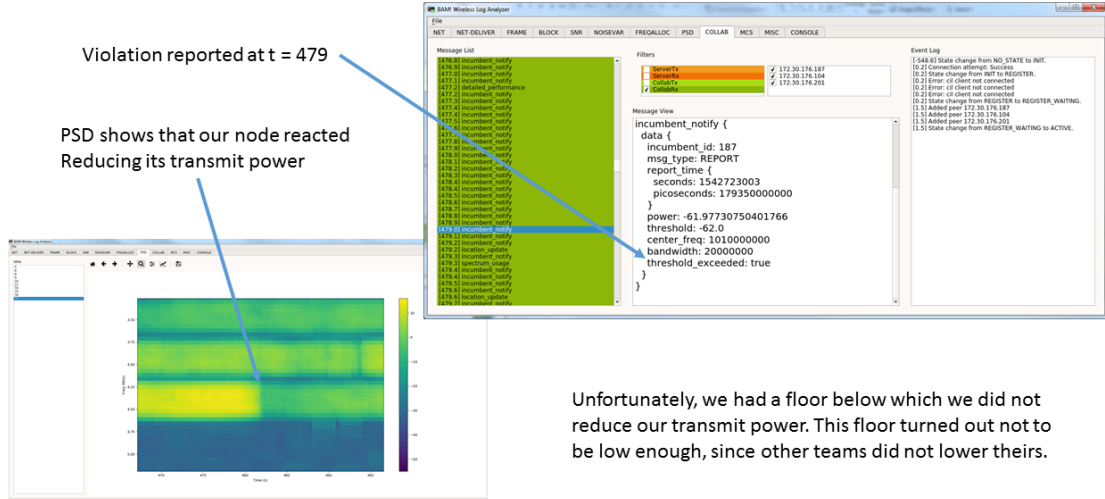


Figure 13: Captures from our GUI showing our protection of the passive incumbent.

free play. Colosseum was usually too busy to schedule large jobs, so our ones would sit in the queue for very long periods. Eventually, we had to base a lot of our decisions on the results that we saw with 30 nodes, and it seems like this did not capture the case when 20 more nodes were in play. We did benefit greatly from the additional opportunity, as we could test extensively through 50 node payroll reservations, and investigate the challenges that our CIRN had in the payroll stage.

## 5 Phase 3 Planned Changes

In this section, we discuss the main planned changes in Phase 3 for the decision making algorithm. Our main focus in this final phase is on augmenting various parts of the decision making algorithm with intelligent capabilities for autonomous adaptation to complex environments. We will place particular emphasis on:

1. Channel allocation.
2. Flow scheduling.
3. MCS selection.
4. Context understanding (e.g., peer and scenario identification).

Further, the above changes will take into account the new scoring mechanism of Phase 3. We first explain the planned framework for incorporating a data-driven machine learning framework, and then illustrate how the framework would be applied to each of the above items.

### 5.1 Machine Learning Framework and Data Generation

The approach we plan to follow is to build on refined versions of our current solutions to provide candidate solutions for each of the considered tasks below. A machine learning model is then trained to learn the solution that best serves the scoring criterion in each situation. The model is aided by data generated using Colosseum reservations and freeplay matches.

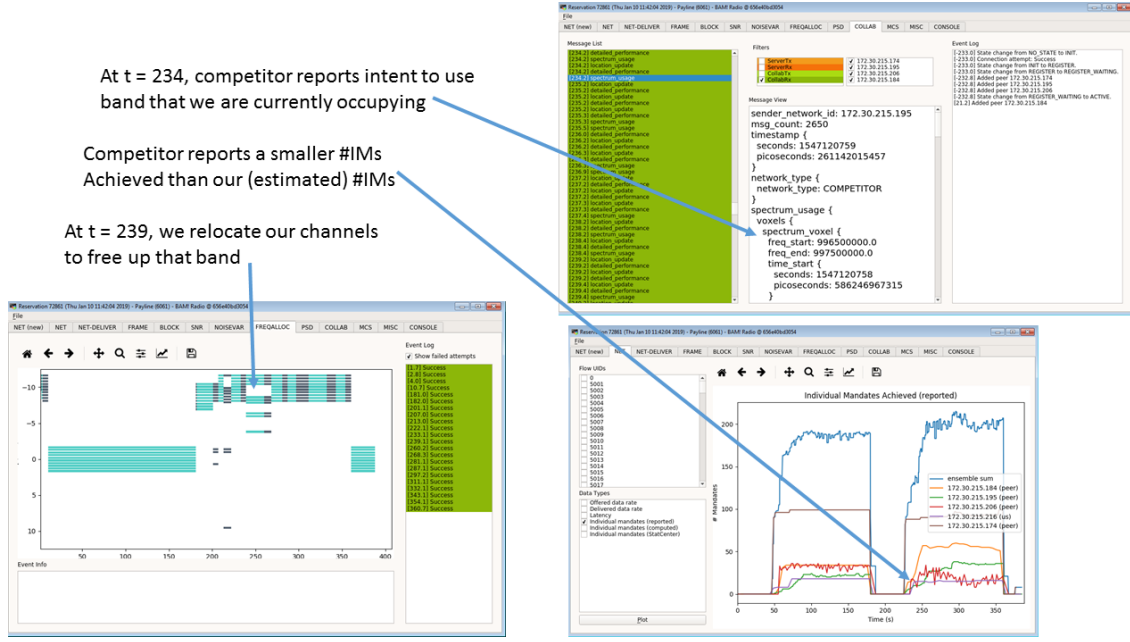


Figure 14: Captures from our GUI showing our collaboration to maximize the minimum score across the ensemble.

### 5.1.1 Deep Learning Model

The model's input is given by the candidate solutions as well as environment features, that are specified below for each of the tasks. The *class label* is a discrete score that reflects the contribution of the decision made to the match score. The sample resolution is the time between applying the candidate solution and measuring the discrete score. We plan to test several state of the art deep neural network architectures that have had successes in related applications (e.g., CNN, CLDNN with various numbers of LSTM layers, ResNet with various choices for the shortcut connections).

### 5.1.2 Dataset Generation

The dataset will be generated from 30-minute matches run over Colosseum. If the sample resolution is one second, then we can collect 1800 samples for each match. By running around 50-55 matches, we obtain approximately 100k samples, which would be an appropriate dataset for training and testing a deep neural network.

## 5.2 Augmenting Current Solutions

In Sections 5.2.1 - 5.2.4, we start by explaining the variables that lead to the multiple candidate solutions that contribute to the input of the deep learning model. We then identify the important features that will augment the candidate solutions to form the input vector.

### 5.2.1 Channel Allocation

In order to generate multiple candidate solutions for the proposed deep learning framework, note that the value function shown in Algorithm 3 is based on a heuristic. Further, the employed interference maps need to incorporate information from CIL messages as well as PSD measurements. Varying these two variables will be used to generate multiple sound solutions. The features accompanying the candidate solution to form the input to the deep learning model will be the PSD measurements, other

teams’ occupied and requested channels, current score and the Phase 3 scoring threshold, as well as the Colosseum scenario (see Section 5.2.4) and current stage if the scenario has different stages.

Further, in order to generate candidate solutions, We will experiment with changing our channel allocation strategy by incorporating more advanced learning-based spectrum sharing policies that allow to learn spatio-temporal correlations and dynamics in spectrum occupancy and exploit them to improve the overall spectrum utilization. Specifically, we are envisioning the use of approximate dynamic programming techniques along with reinforcement learning to allow our CIRN to more effectively exploit the spectrum resources available and quickly adapt to them, and to constantly improve and adjust these policies based on ongoing interactions with the environment.

### 5.2.2 Flow Scheduling

We envision a change in our flow scheduling algorithm based on the anticipated change in scoring in Phase 3. However, the structure of the adaptive algorithm need not change. As illustrated in Section 3.3.2, our flow scheduling algorithm in Phase 2 relied on the fact the Phase 2 scoring mechanism assigned equal value to all flows. Delivering a Command and Control (C2) file had the same value as delivering a large file. While the mechanism would remain the same, there are multiple ways now to incorporate the new Phase 3 scoring mechanism, while searching for the optimal combination of flows.

The important features that should accompany the candidate solutions to form the input of the deep learning model are:

- **Allocated Bandwidth:** Since the feasibility of delivering a given combination of flows depends on the available bandwidth.
- **MCS Choice:** The choice of modulation scheme and coding rate will affect the SINR, and hence, it is important for determining the optimal combination of flows.
- **Current Estimated Score:** The current score is an important factor in determining our objectives, and hence, the target combination of flows that would best serve this objective. Particular emphasis here would be on whether the new *threshold* is passed or not.

### 5.2.3 MCS Selection

The MCS adaptation algorithm illustrated in Algorithm 1 relies on two important parameters that are determined in heuristic ways. Namely, the estimate of the noise variance and the block error rate bound. Variations to these heuristics will be used to generate multiple candidate solutions. Further, we did not take into account other environment parameters like the bandwidth allocated. Hence, one possibility would be to investigate designing other candidate solutions that take these parameters into account.

The important features fed into a deep learning model to learn the optimal MCS selection in each setting are:

- **Block Error Rate:** This is a reasonable choice, since we already tested successful heuristics based on the block error rate.
- **Node Locations and Transmit Powers:** Our own and other teams’ node locations and transmit powers are important to determine the right choice of MCS.
- **Scheduled Flows:** The type of flows being delivered should be used to determine the right MCS choice.

- **Score:** Our current score is an important factor. For example, with the new Phase 3 scoring, if the ensemble passed the *threshold*, we should use more aggressive MCS choices.

#### 5.2.4 Context Understanding

Here, there are two key intelligent capabilities that we aim to develop, to enhance the performance of the learning process of the above mentioned tasks. Namely,

1. Creating Signatures for Other Teams.
2. Colosseum Scenario Identification.

For the first task, there are 14 other finalist teams in Phase 3. We suspect that shallow learning methods could be effective here, since we have experience with observing logs for the competitor teams, and may be able to craft features that guide the model to identify the other ensemble teams. Since we never know the identity of other teams while generating the dataset, the problem becomes an unsupervised learning problem, where we want to cluster the input data into 14 regions. One approach we will give priority to testing is a graph-based approach, where each feature vector is represented as a vertex on the graph, and a graph clustering method (e.g., spectral clustering) is employed.

We believe that the scenario identification task could be simpler, because we know the scenario descriptions, and hence, could design conditions to identify certain scenarios. However, learning could prove to be useful to distinguish between scenarios having similar features, as well as to learn the behavior of active incumbents and identify the presence of jamming agents. Important features for this task are the available bandwidth, time since the start of the match, own node locations, and offered traffic.

## 6 Conclusions

In this paper, we provided a description of the key aspects of the BAM! Wireless CIRN design during Phase 2 of the DARPA SC2 Challenge. Our current solution provides a robust and agile framework, that we can build on to add more sophisticated features during Phase 3. In particular, we expect that employing the proposed machine learning framework could particularly help find the optimal heuristic solution in various settings. Further, using multiple antennas and enhancing the adaptive MCS algorithm through incorporating information about the transmit power level and extensive testing, as well as noise whitening for narrowband channels would greatly enhance the physical layer. Also, using non-binary routing tables as well as incorporating stream ARQ would improve the reliability of flow delivery while meeting requirements. Finally, added features in Phase 3 like the presence of active incumbents, varying scoring policies for different flows and different times of the match would also require changes to multiple parts of our design.

## References

- [1] Team BAM! Wireless, “DARPA SC2: Functioning Subsystem Design Report”, 2018.
- [2] M. Shreedhar and G. Varghese, “Efficient fair queuing using deficit round-robin,” *IEEE/ACM Transactions on Networking*, vol. 4, no. 3, pp. 375–385, Jun 1996.