Suggestions for Improvements to the BAM-Wireless radio node design for SC2 Phase 3

Bharath Keshavamurthy

March 2019

1 Work in Progress

- *Updates to the source code in order to incorporate new scoring rules in Phase-3*: Differentiated Flows and Bonus Thresholds

- *Throughput-Optimal Cross-Layer Design*: Formulate a Network Utility Maximization (NUM) problem and solve it using Lagrangian Duality methods and heuristic algorithms in order to arrive at throughput-optimal solutions for MCS Adaptation, Channel Allocation, Routing, and Weighted-Flow Scheduling
  - A Utility Maximization Problem with constraints corresponding to protocols in each layer of the cognitive radio node's protocol stack where the utility function can be a simple log-utility function as seen in proportional fairness problems, is formulated.
  - Constraints include non-interference with the incumbent, minimum throughput requirements (for each allotted flow), flow-routing constraints (node-balance, non-negative rates, and capacity constraints), non-interference among secondary nodes, and constraints for preventing cyclic paths in routes
  - Decompose the NUM problem into sub-problems by removing the coupling among constraints and tackle each sub-problem separately
  - Solve these problems using sub-gradient methods with gradient-projection - some solutions are discussed in [1] and [2]. They include,
    - Weighted Flow Scheduling - Back-pressure scheduler (assign bandwidth to a flow with the highest utility metric in terms its point value and queue differential measure)
    - Routing - Choose the most stable routes instead of simple minimum-hop routing
    - Channel Allocation - Each node calculates, for each of its outgoing links, the belief that a channel is idle/free in its interference region. Additional requirements will be included for satisfying

the scenario gates - non-interference with the incumbent and access only to the allowed portion of the spectrum. The nodes will follow CSMA in the MAC layer with an exponential back-off procedure based on this channel availability metric and the queue differential measure. The higher the availability of the channel or the higher the back-pressure on the link, the more aggressive is the CSMA back-off policy.

- Assuming a Markovian Correlation among channels and across time wherein the model parameters are learnt over time using an online Parameter Estimation Algorithm, we can formulate a POMDP that provides the channel availability metrics outlined in the previous point. Since the belief space is going to be huge and the observation space continuous, we can use Approximate Value Iteration Algorithms like the PBVI (Point-Based Value Iteration) Algorithm or the PERSEUS Algorithm (Randomized Point-Based Value Iteration) in order to extract the channel availability metrics during the course of interaction of the POMDP agent with the radio environment and combine it with the queue differential measure to come up with the optimal policy.

2 Possible Extensions

There is the problem of System Dynamics involved in the POMDP formulation. Since there are a lot of moving parts to the radio environment we're operating in, recent research detailed in [3] and [4] suggest that Re-Learning OR Re-Training gives us significantly better performance than our current approach of assuming orthogonality among channels and operating a reactive strategy.
For instance, a proactive strategy with re-training is shown to be an effective solution as opposed to a Reactive Whittle-Index based strategy in [3]. We can take two approaches to re-training:

- We can develop extensions to the work outlined in [4] in order to re-learn the most relevant belief states and then use those in our Approximate Value Iteration Algorithm to solve for an optimal policy.
    - We can re-sample the set of reachable belief points using our most recent policy when the accumulated reward experiences a significant drop.
    - We can then use this new set of reachable belief points to solve for a new optimal policy using the PERSEUS algorithm (”Backup” until all belief points in the reachable set have been sampled and Update the Value Function for these beliefs based on the chosen policy tree vector).
- Another possible approach would be to employ Adaptive Deep Q-Networks. Adaptive DQNs turn out to be great tools to solve for an optimal policy in highly-dynamic environments where the system statistics are unknown.

- Reference [3] uses the Deep Q-Learning with Experience Replay Algorithm to design the DQN.
    - * Design: 2-layer Neural Network with 200 neurons and a ReLU activation function at each neuron
    - * An ϵ-greedy policy is employed to select actions (channel combinations) in a given state. The interaction record - *(state, action, observation, reward, next-state)* is stored in a "Replay Memory" and a random set of these historic records are used to compute the loss function.
    - * The weights of the DQN are updated using the stochastic gradient-descent algorithm
- This DQN is used in conjunction with a re-training policy which involves evaluation of the accumulated reward of the current policy and simply comparing it with a threshold in order to trigger re-training to find a new good policy.

3 References

1. "Cross-Layer Optimization and Protocol Analysis for Cognitive Ad Hoc Communications"

2. "Throughput-Optimal Cross-Layer Design for Cognitive Radio Ad Hoc Networks"

3. "Deep Reinforcement Learning for Dynamic Multi-Channel Access in Wireless Networks"

4. "Perseus: Randomized Point-based Value Iteration for POMDPs"

5. A tutorial on cross-layer optimization in wireless networks"

6. Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures"