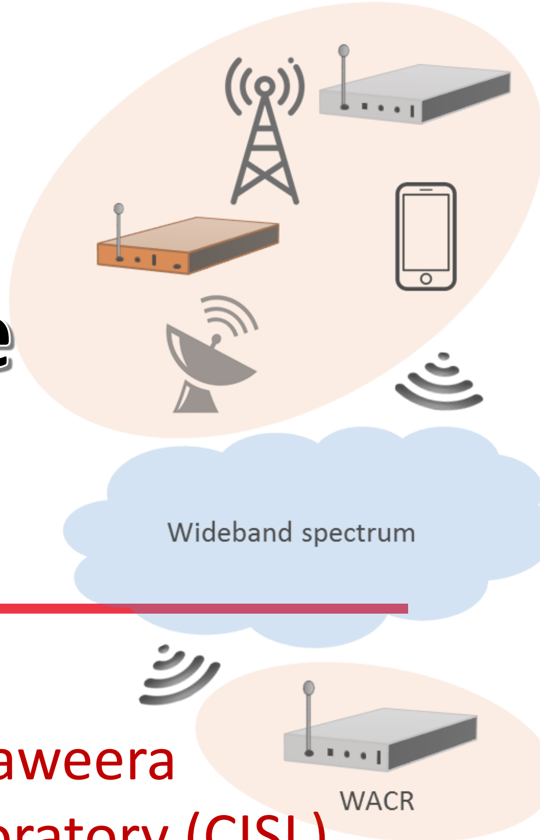
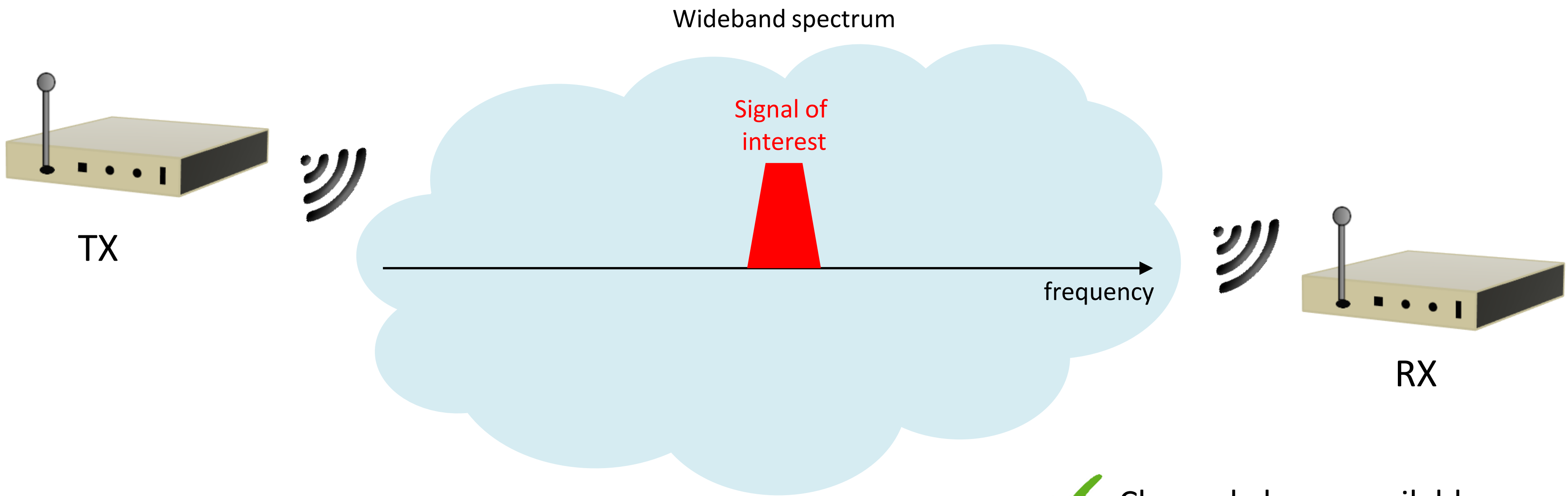
Robust Deep Reinforcement Learning for Interference Avoidance in Wideband Spectrum



Mohamed A. Aref and Sudharman K. Jayaweera
Communication and Information Sciences Laboratory (CISL)
Department of ECE, University of New Mexico
{maref, jayaweera}@unm.edu

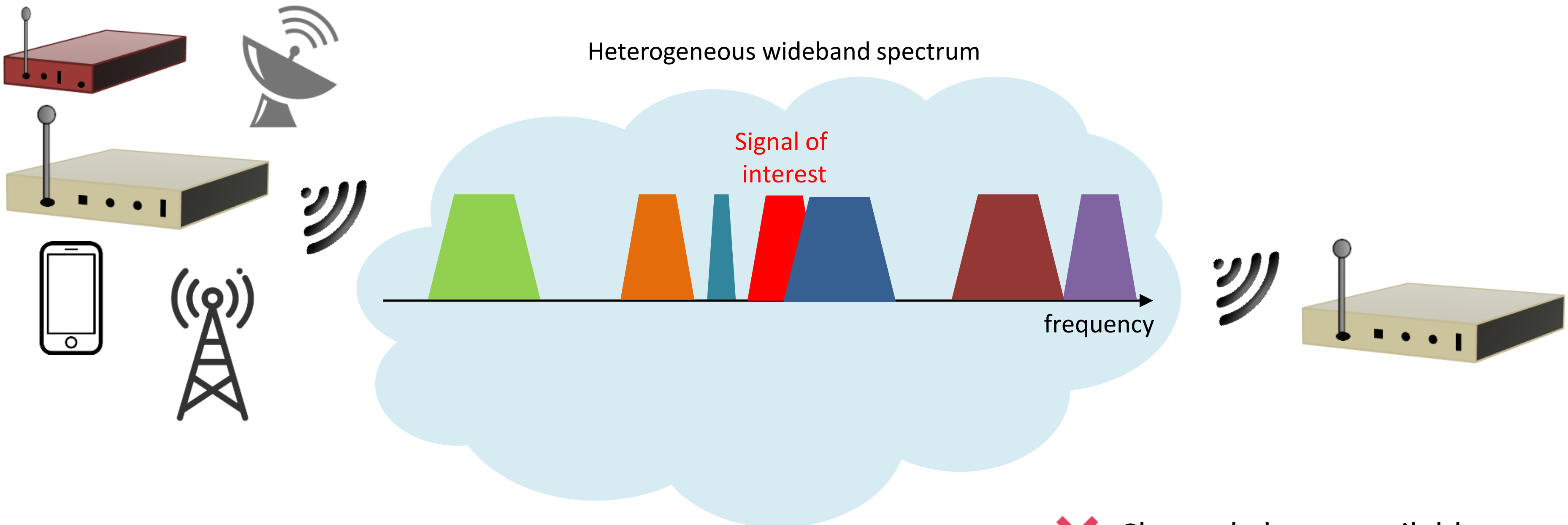
Motivation



- ✓ Channel always available
- ✓ No interference
- ✓ No malicious attacks
- ✓ Receive signal successfully

Ideal World !

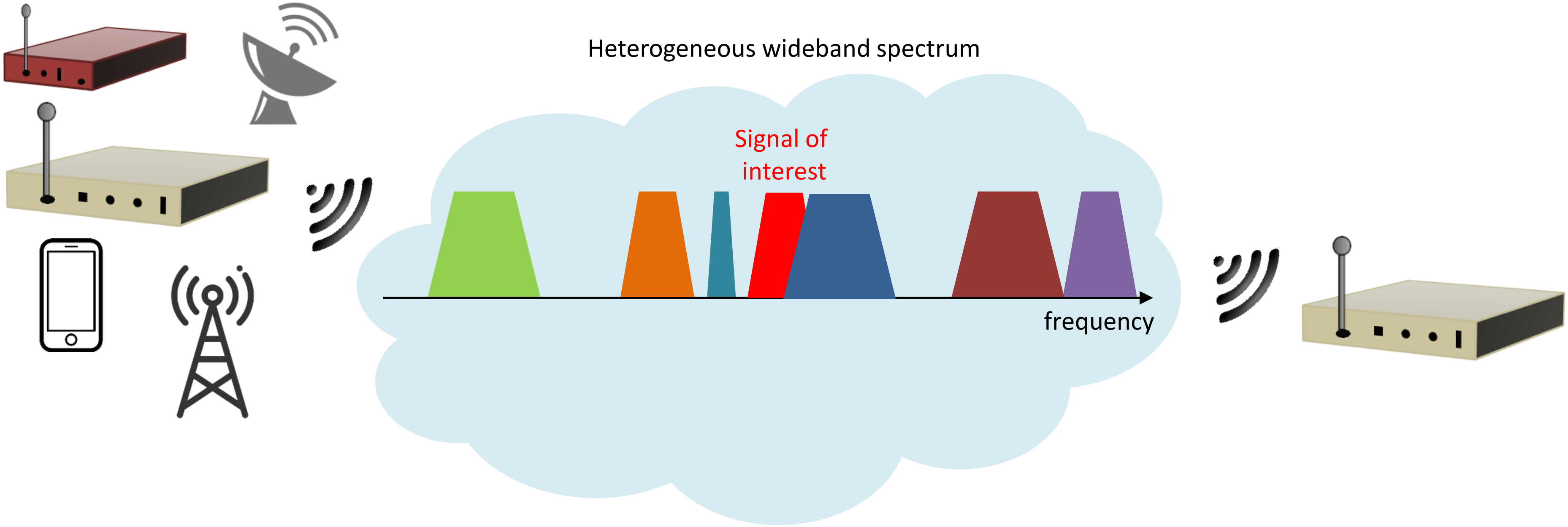
Motivation



❌ Channel always available
❌ No interference
❌ No malicious attacks
❌ Receive signal successfully

How to find spectrum opportunities in such heterogeneous RF environment?

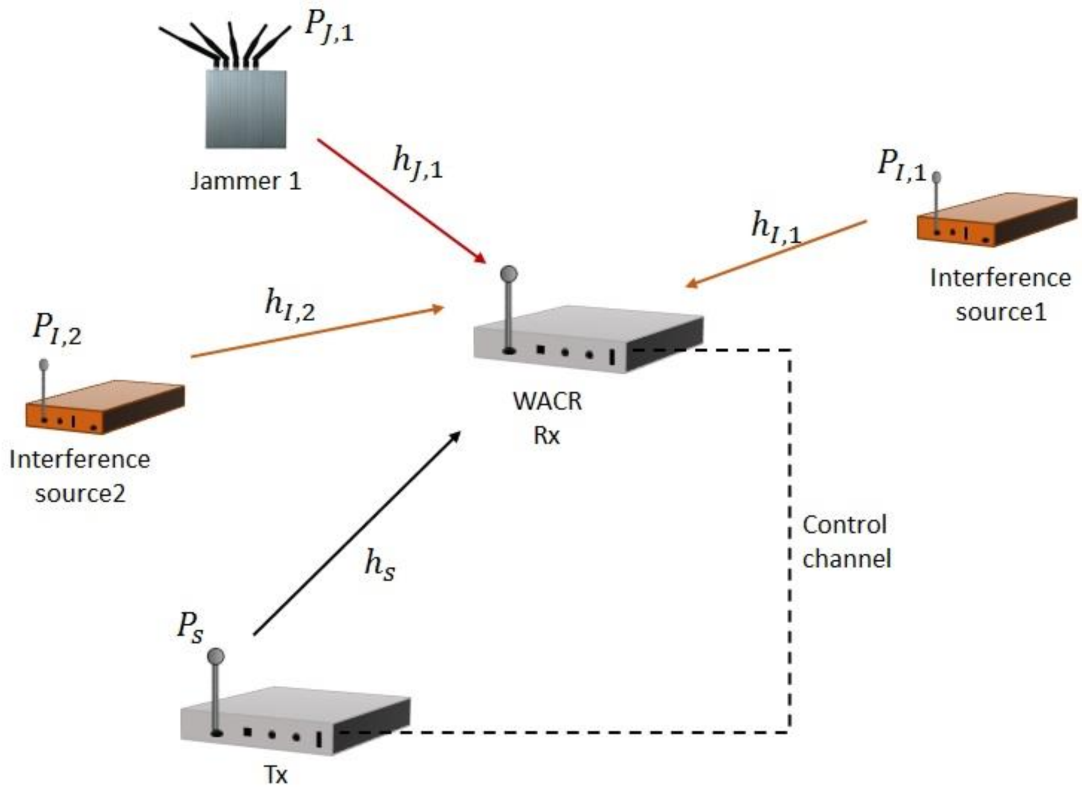
Motivation



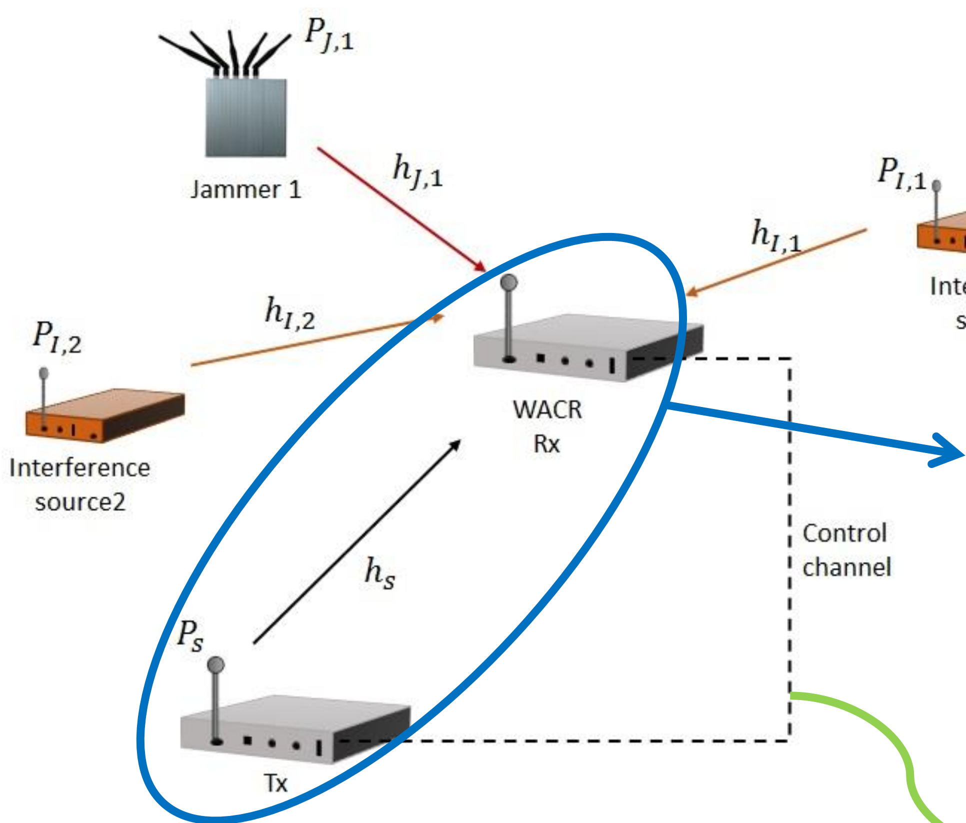
Properties of proposed technique

- Ability to learn efficient channel-selection policy to avoid interference, jamming and any other harmful signals
- Ability to work in a partially-observable RF environment
- Rapid reconfiguration to tackle sudden changes in the RF environment
- Low computational complexity

System Model



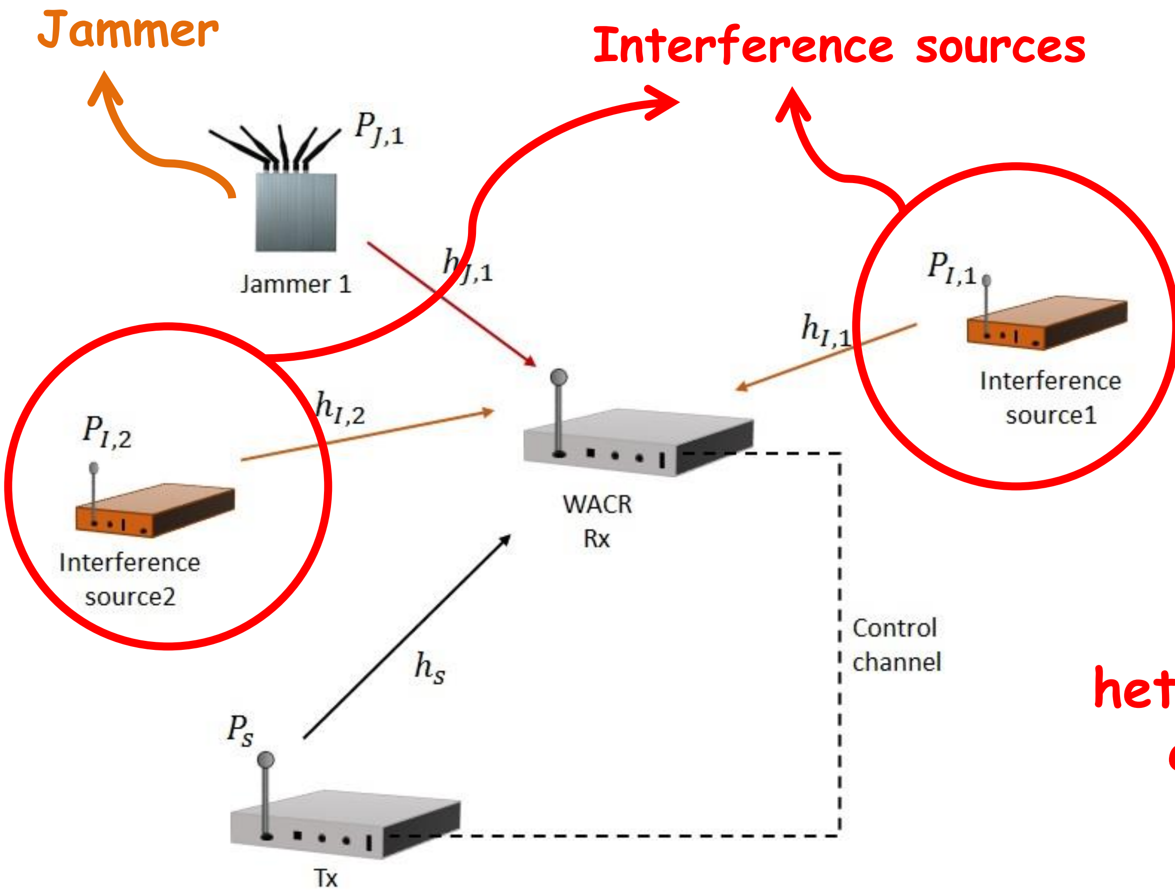
System Model



- Link of interest
- Rx node has cognitive capabilities

- Secured control channel between Tx and Rx

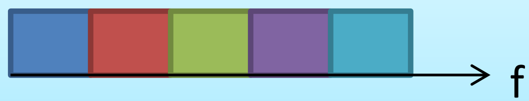
System Model



heterogeneous RF environment

System Model

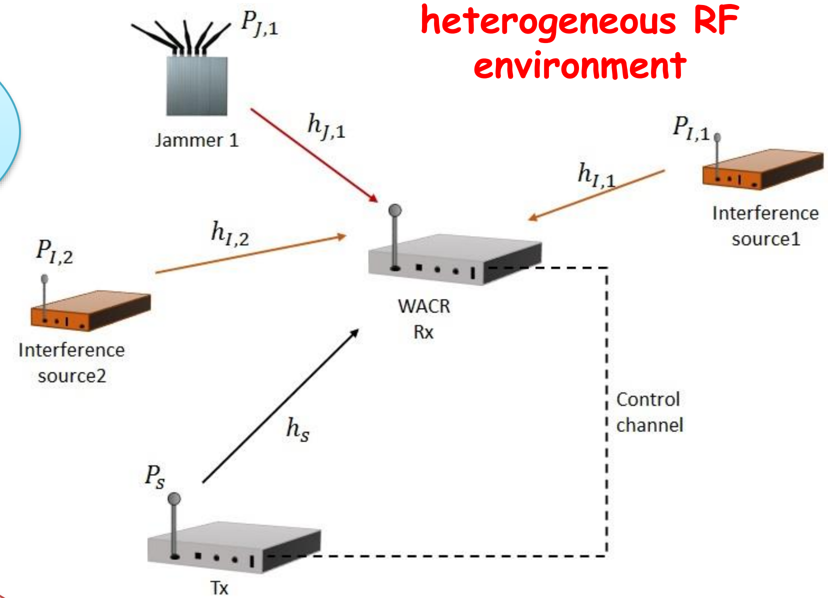
The RF spectrum of interest



N frequency channels

The WACR needs to choose a frequency channel that achieves the highest possible **SINR** value.

heterogeneous RF environment



System Model

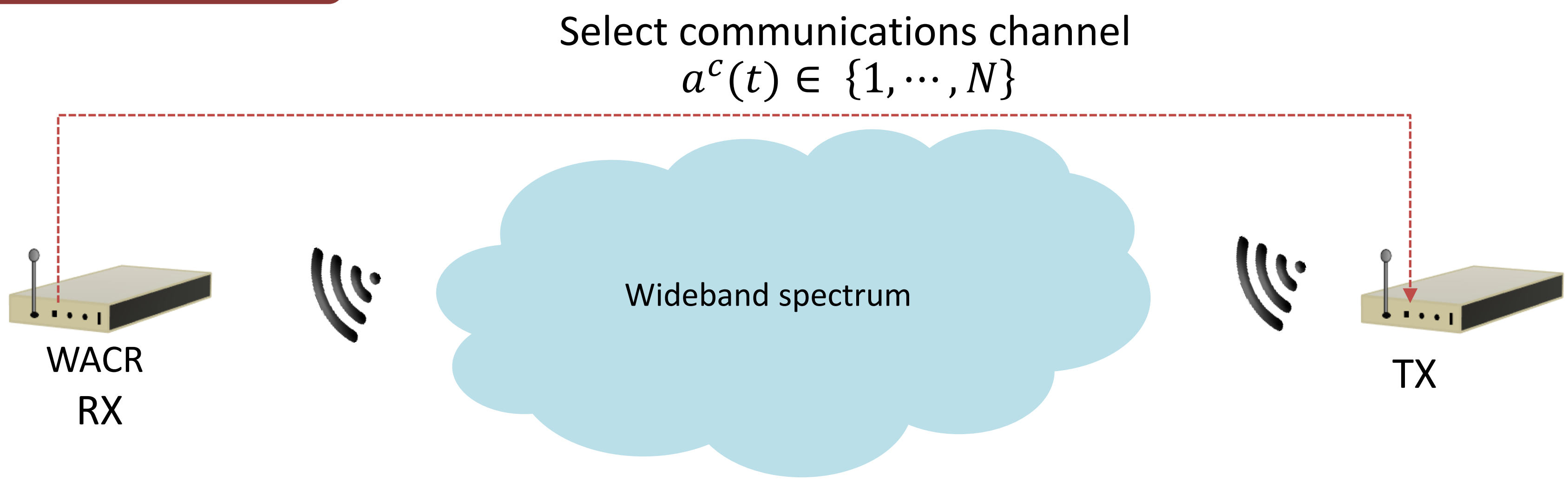
The received SINR of the WACR in channel $a^c(t)$ at time t can be expressed as

$$\boxed{\mu_{a^c(t)} = \frac{h_s P_s}{\sigma^2 + \sum_i h_{I,i} P_{I,i} + \sum_j h_{J,j} P_{J,j}}} \quad (1)$$

- P_s : The transmitted power for the signal of interest.
- h_s : The channel power gain from Tx to WACR (Rx).
- $P_{I,i}$: The transmitted for the signal of interference source *i*.
- $h_{I,i}$: The channel power gain from interference source *i* to WACR.
- $P_{J,j}$: The transmitted power for the signal of jammer *j*.
- $h_{I,i}$: The channel power gain from jammer *j* to WACR.
- σ^2 : The receiver noise power, assuming AWGN.

System Model

Communications



- Estimate SINR $\mu_{a^c(t)}$
- The function $g(.)$ indicates the success of the communications

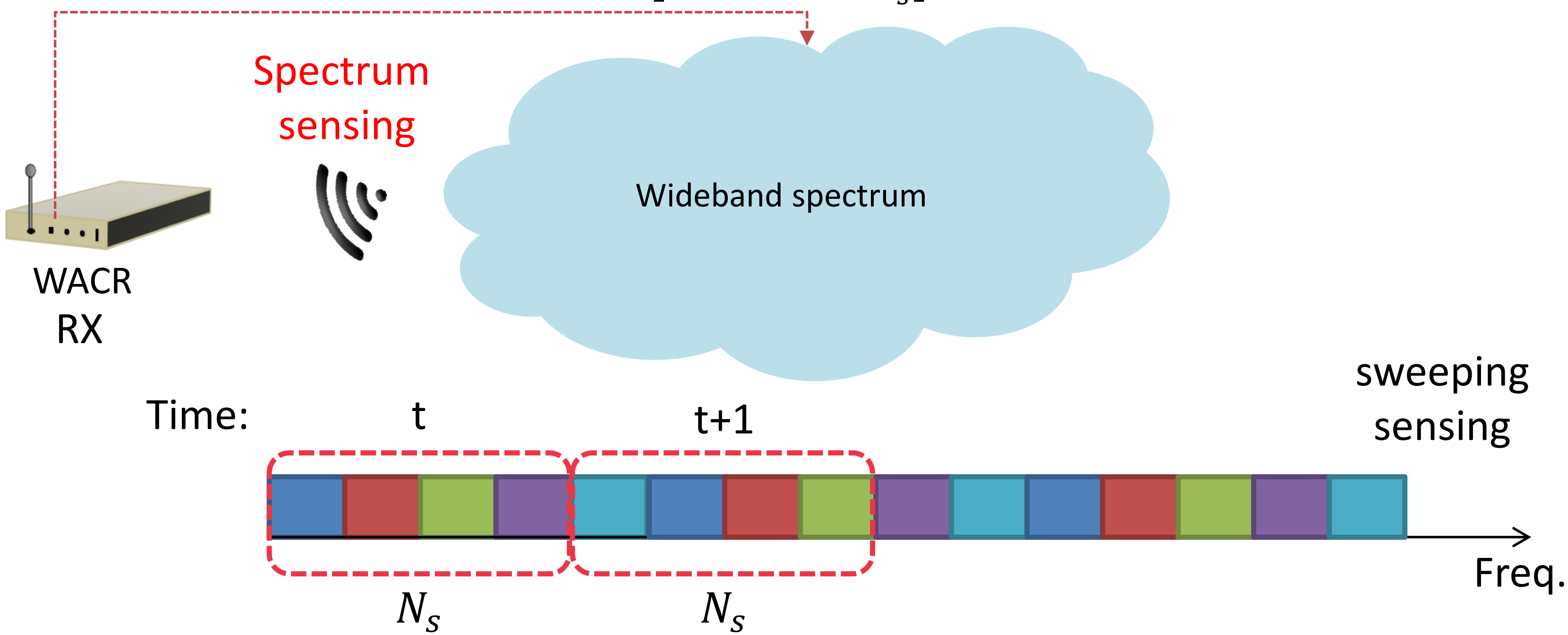
$$g(\mu_{a^c(t)}) = \begin{cases} \lambda, \; if \; \mu_{a^c(t)} > \mu_{th} \\ -\lambda, if \; \mu_{a^c(t)} \leq \mu_{th} \end{cases}$$

System Model

Sensing

Select sensing channel

$$\boldsymbol{a}^s(t) = \left[a_1^s(t), \cdots, a_{N_s}^s\right], \text{ where } N_s < N$$



- The function $f(.)$ indicates the availability of the sensing channels

$$f(v_{a_i^s(t)}) = \begin{cases} -\lambda, & if\ v_{a_i^s(t)} > v_{th} \\ \lambda, & if\ v_{a_i^s(t)} \leq v_{th} \end{cases}$$

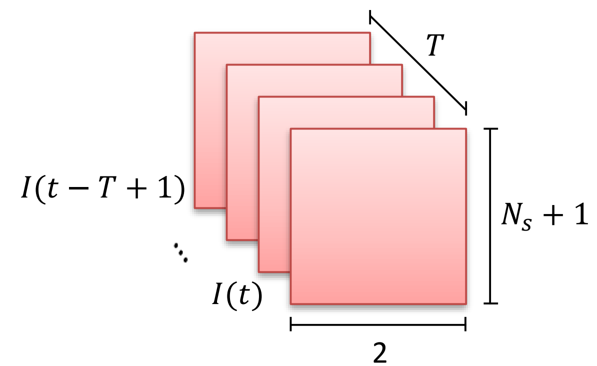
System Model

Using the information from both communications and sensing

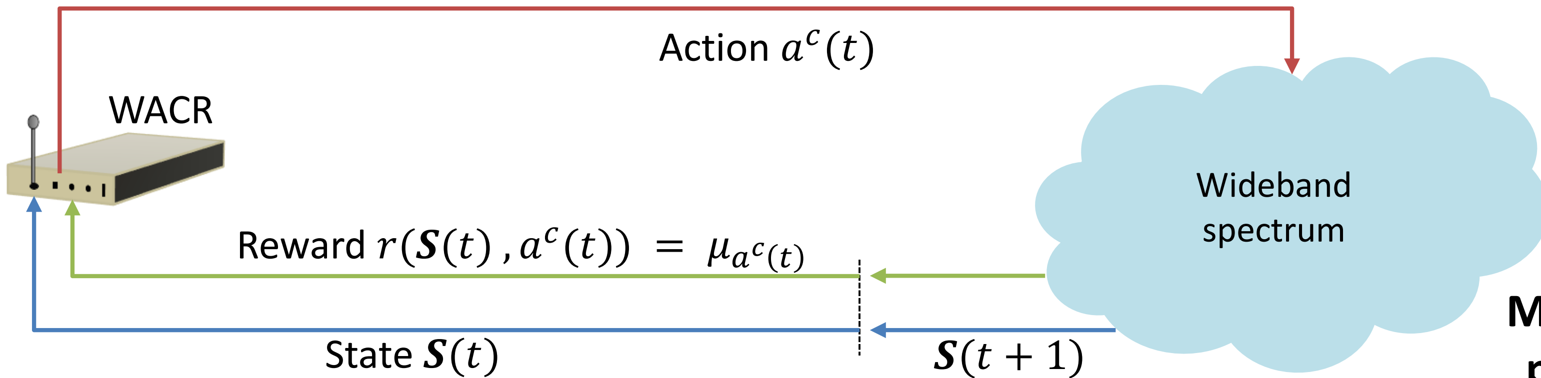
$$\mathbf{I}(t) = \begin{bmatrix} a^c(t-1) & g(\mu_{a^c(t-1)}) \\ a^s_1(t-1) & f(\nu_{a^s_1(t-1)}) \\ \vdots & \vdots \\ a^s_{N_s}(t-1) & f(\nu_{a^s_{N_s}(t-1)}) \end{bmatrix}$$

The state $\boldsymbol{S}(t)$ is made of T successive indication matrices up to time t

$\boldsymbol{S}(t) =$



Proposed Cognitive Engine



Markov decision process (MDP)

Using reinforcement learning (e.g. Q-learning)

Problems

- The number of possible states can become extremely large even for few frequency channels and few time slots.
- The learning speed would be an obstacle to work efficiently in real-time.

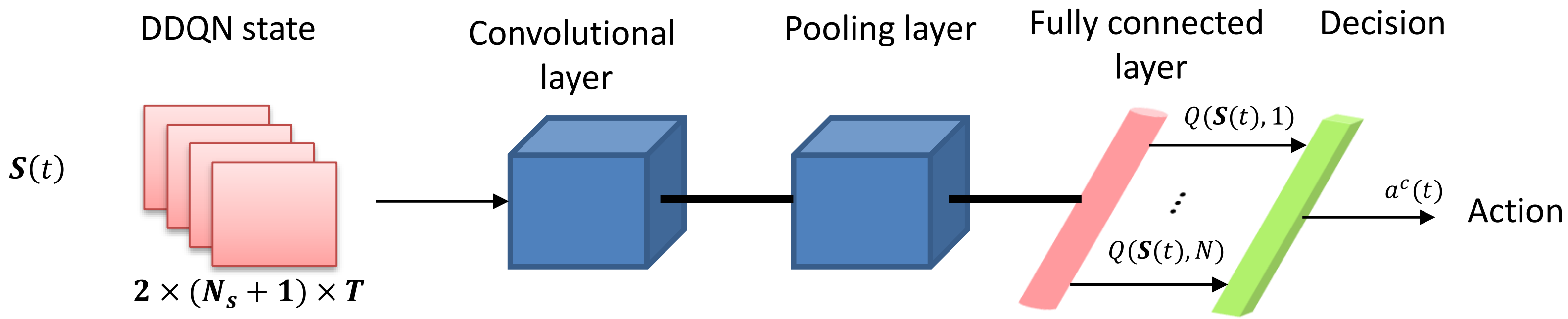
Solution?

Use Deep reinforcement learning

Double Deep Q-network (DDQN)

Combine reinforcement learning with convolutional neural network (CNN)

Proposed DDQN Algorithm



- For a given state $\boldsymbol{S}(t)$, the CNN is used to estimate the Q- function $Q(\boldsymbol{S}(t)\,, a^c(t))$ for each possible action $a^c(t)\ \in$ *{1,…,N}.*

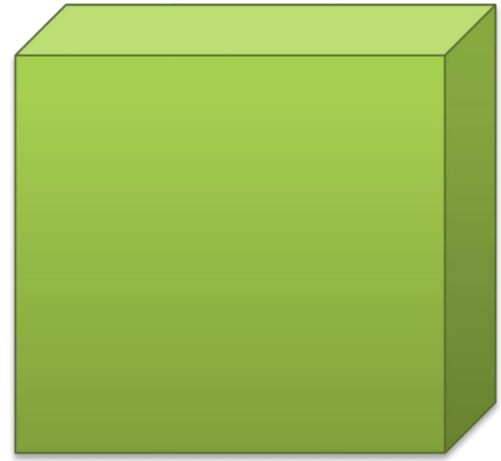
- The WACR selects an action $a^c(t)$ that represents the index of the communications channel at time $t + 1$

$$\boxed{a^c(t) = \begin{cases} \arg\max\limits_{\acute{a} \in \mathcal{A}^c} Q(\mathbf{S}(t), \acute{a}; \theta(t)) & \text{with probability } 1-\epsilon \\ \sim U(\mathcal{A}^c) & \text{with probability } \epsilon, \end{cases}}$$

Proposed DDQN Algorithm

Experience replay

Data set
$\mathcal{D}(t)$



$$x(t) = \big(\boldsymbol{S}(t), a^c(t), \mu_{a^c(t)}, \boldsymbol{S}(t+1)\big) \xrightarrow{\text{store experience}}$$

$$x(k) \sim U(\mathcal{D}(t)) \xleftarrow[1 \leq k \leq t]{\text{draw randomly}}$$

- Break temporal correlation between training examples.
- Use stochastic gradient descent (SGD) to update network weights $\theta(t)$

Loss function $\Rightarrow$

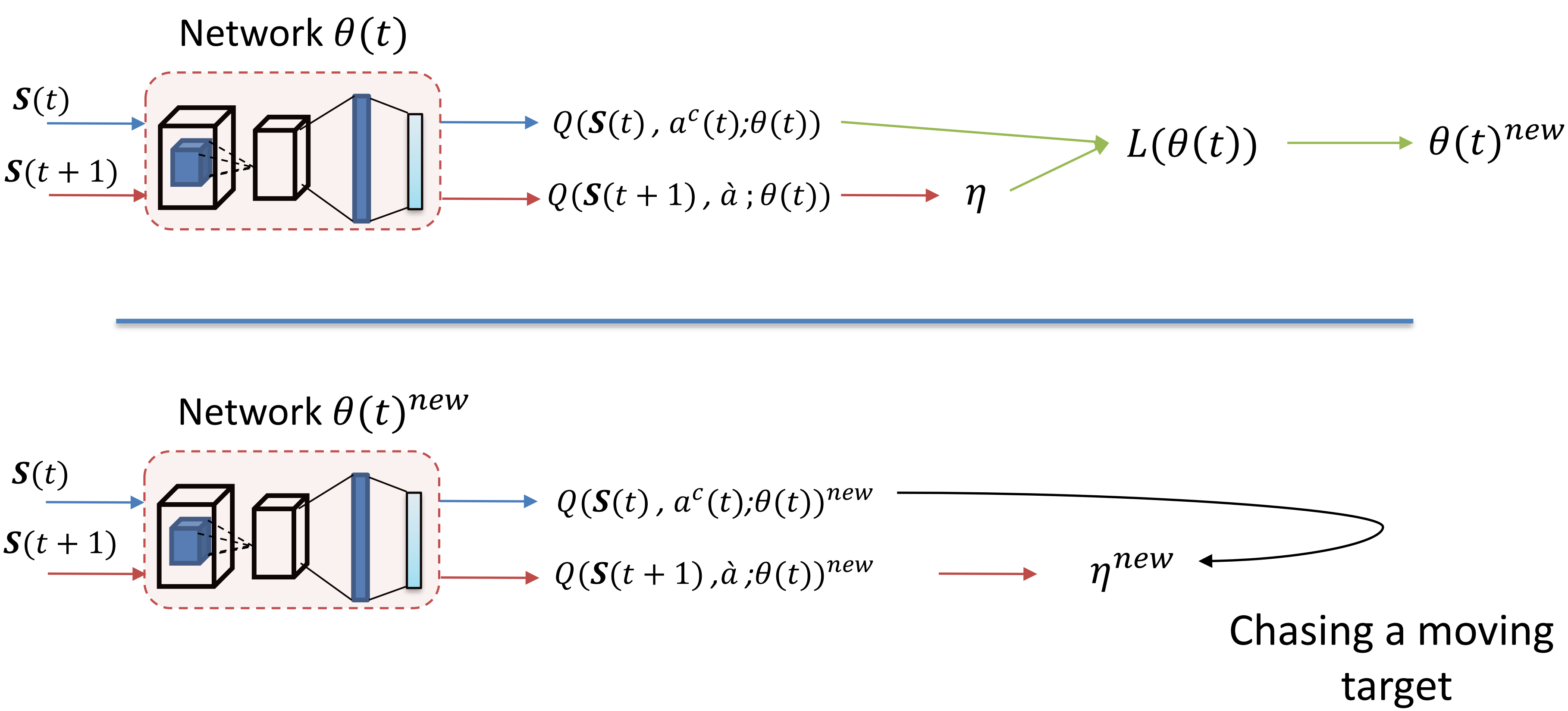
$$\boxed{L(\theta(t)) = \mathbb{E}_{x(k)\sim U(D(t))}[(\eta - Q(\mathbf{S}(t), a^c(t); \theta(t))^2]}$$

Target value $\Rightarrow$

$$\boxed{\eta = \mu_{a^c(t)} + \gamma \max_{\grave{a}} Q(S(t), \grave{a}; \theta(t))}$$

Proposed DDQN Algorithm

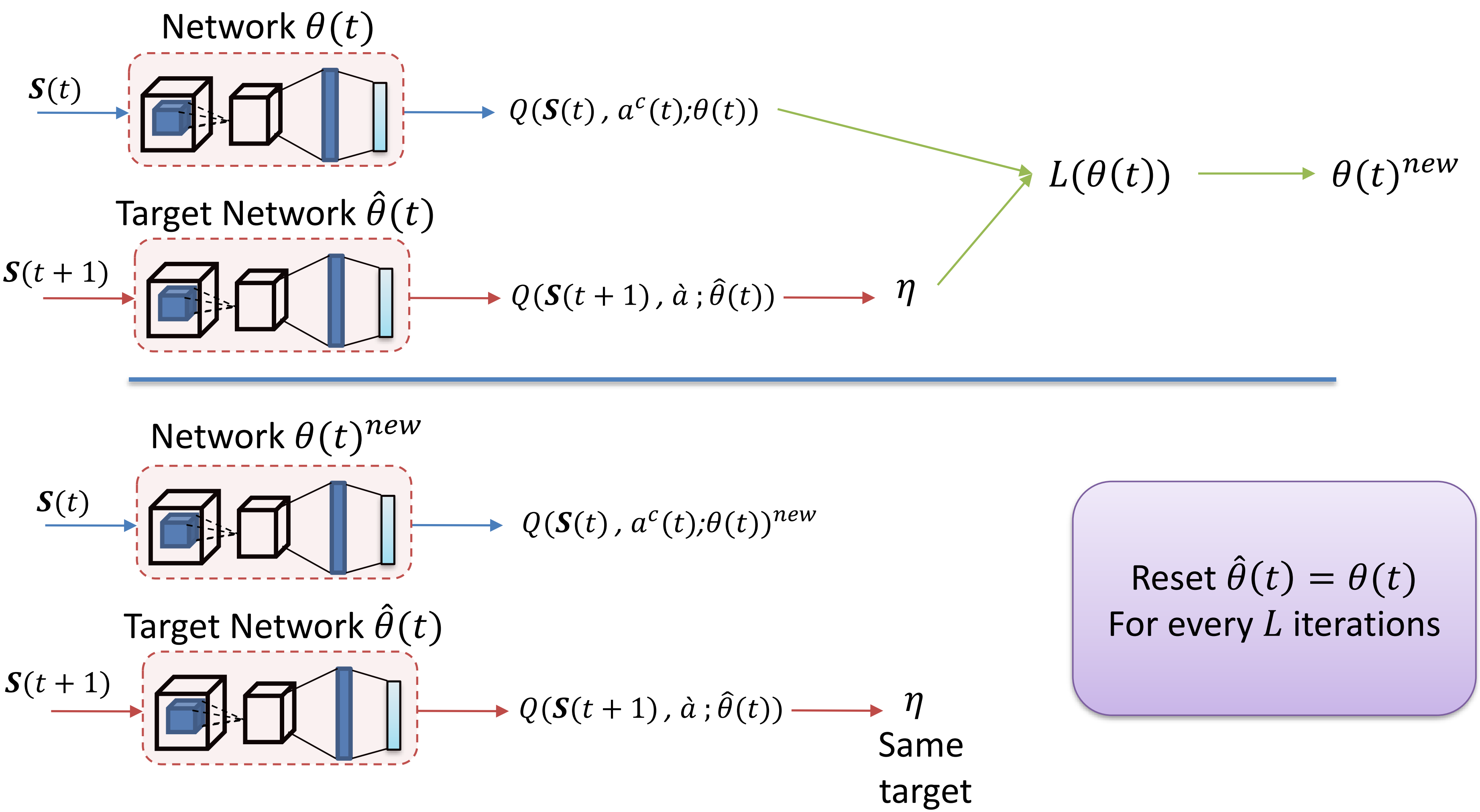
Target Network



Proposed DDQN Algorithm

Target Network

$$\eta = \mu_{a^c(t)} + \gamma \max_{\grave{a}} Q(S(t), \grave{a}; \hat{\theta}(t))$$

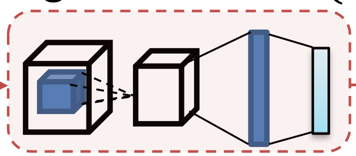


Proposed DDQN Algorithm

DDQN vs DQN

Target Network $\hat{\theta}(t)$

$\boldsymbol{S}(t+1)$



$Q(\boldsymbol{S}(t+1), \grave{a}; \hat{\theta}(t))$

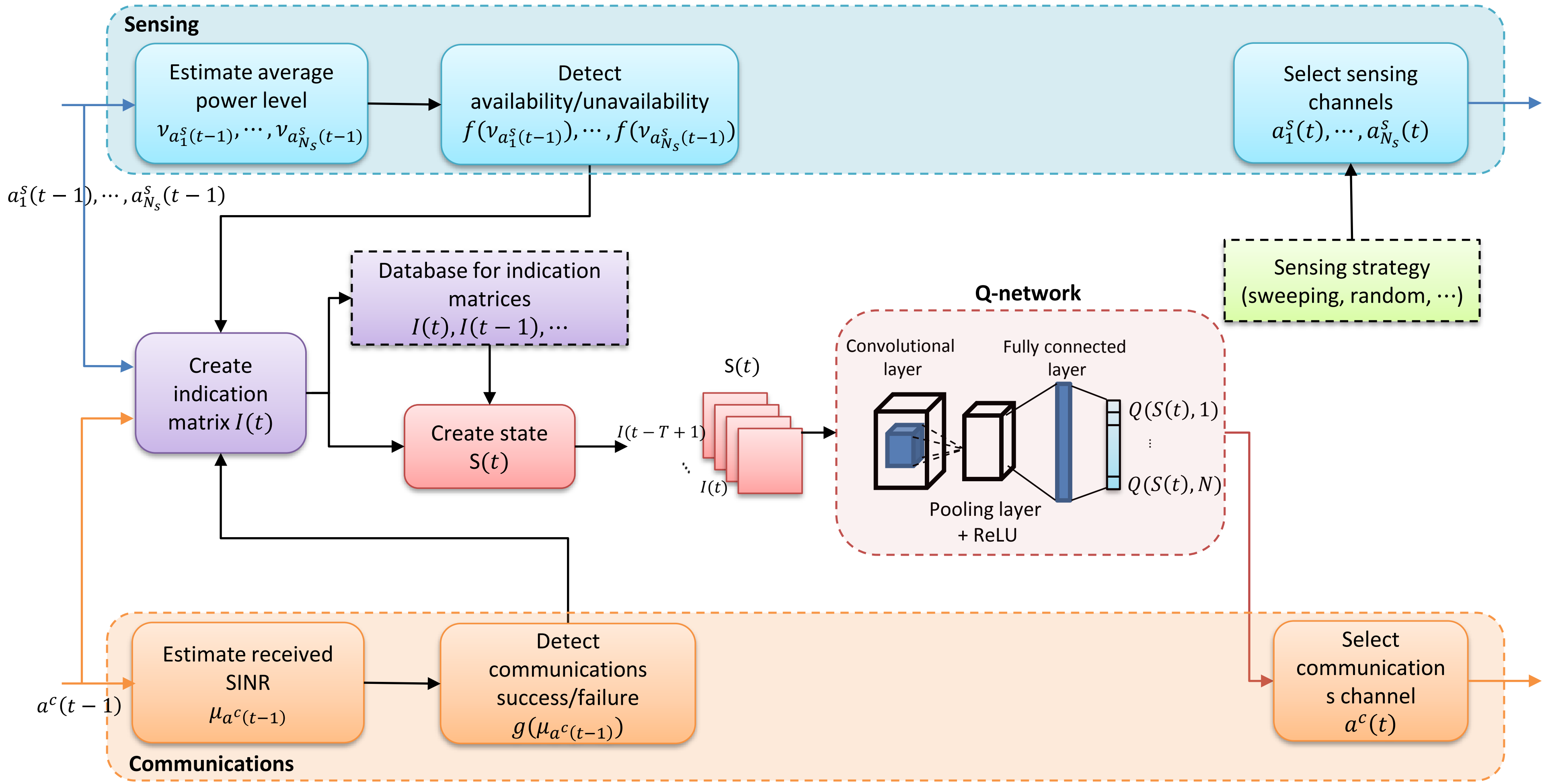
$$\boxed{\eta = \mu_{a^c(t)} + \gamma \max_{\grave{a}} Q(S(t), \grave{a}; \hat{\theta}(t))}$$

Same network to select best action and evaluate optimal Q-value

DDQN decouples the selection and evaluation process

$$\boxed{\eta = \mu_{a^c(t)} + \gamma Q[S(t+1), \underbrace{\arg\max_{\grave{a}} Q(S(t), \grave{a}; \theta(t))}_{\text{Select best action using } \theta(t)}; \hat{\theta}(t)]}$$

Proposed Cognitive Engine



Simulation Results

Proposed 1: The proposed technique in this paper

Comparison

Proposed 2

The proposed technique in **

Index of the current operating channel

Indicator of successful communications over channel a^{t-1}

$$S^t = \begin{bmatrix} \boxed{a^{t-1}} & \boxed{\lambda g(\mu^t_{a^{t-1}})} & 0 & & & \cdots & & & & & & 0 \\ f(c_1^t) & \cdots & f(c_{N_s}^t) & \cdots & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & f(c^{t-1}_{N-N_s+1}) & \cdots & f(c^{t-1}_N) \\ 0 & \cdots & 0 & \cdots & f(c^{t-2}_{N-2N_s+1}) & \cdots & f(c^{t-2}_{N-N_s}) & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

Sensing matrix (W^t)

Q-learning

$$S_Q^t = [a_Q^{t-1}, \lambda g(\mu^t_{a_Q^{t-1}})]$$

Random

The WACR randomly chooses a channel

** M. A. Aref and S. K. Jayaweera, "Spectrum-agile Cognitive Interference Avoidance through Deep Reinforcement Learning", *14th EAI International Conference on Cognitive Radio Oriented Wireless Networks (CROWNCOM'19)*, Poznan, Poland, Jun. 2019.

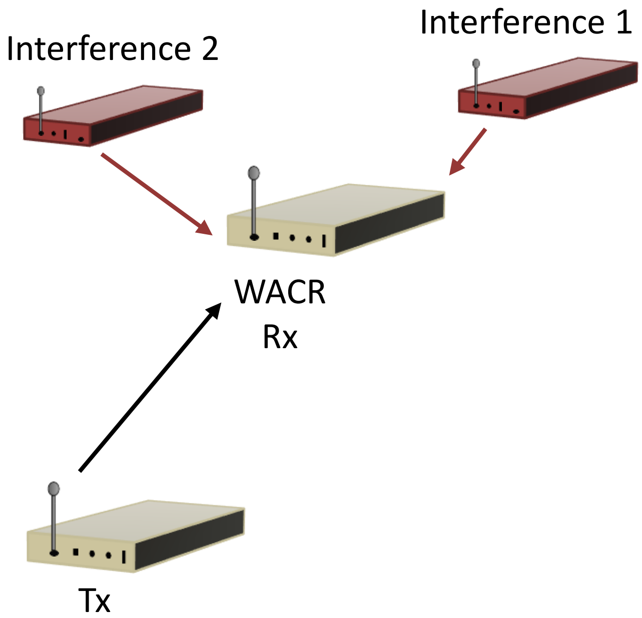
Simulation Results

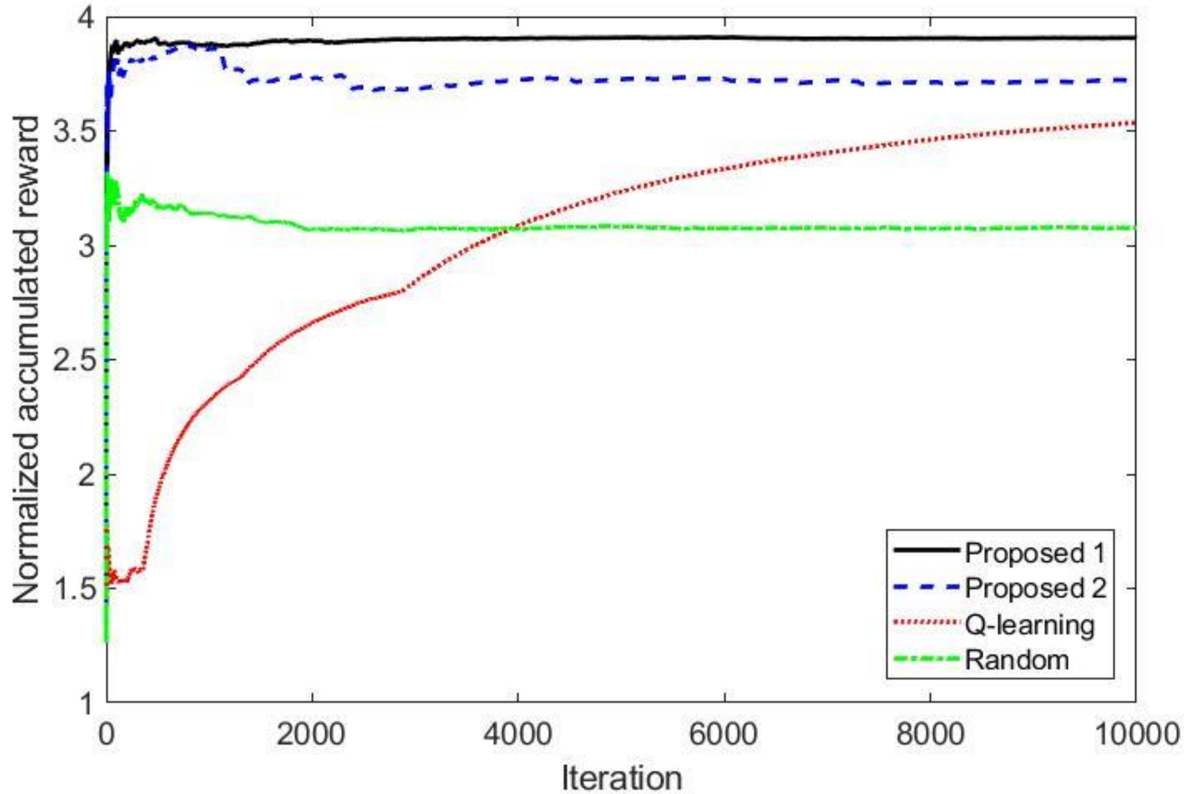
Parameters

Number of channels (N)	6
Number of channels that WACR can sense instantaneously (N_s)	2
Number of time slots in the sensing matrix (T)	3
Transmitted power for the signal of interest (P_s)	5
Channel power gain between Tx and WACR (h_s)	0.8
Noise power of the receiver (σ^2)	1
Thresholds (c_{th} and μ_{th})	2
Number of experience replays for each time slot (K)	5
Learning rate (α)	0.1
Discount factor (γ)	0.4
Exploration rate (ε)	0.1
Weighting factor (λ)	10

Simulation Results

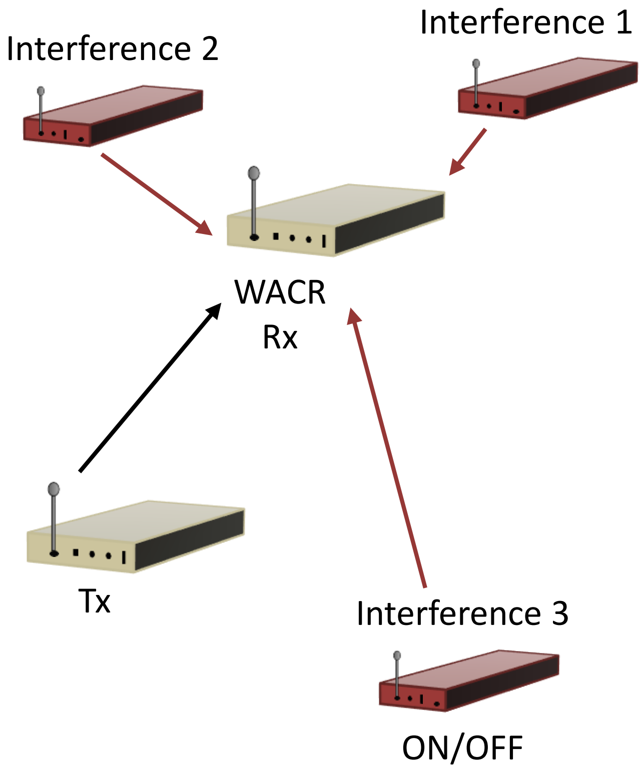
Test case 1:

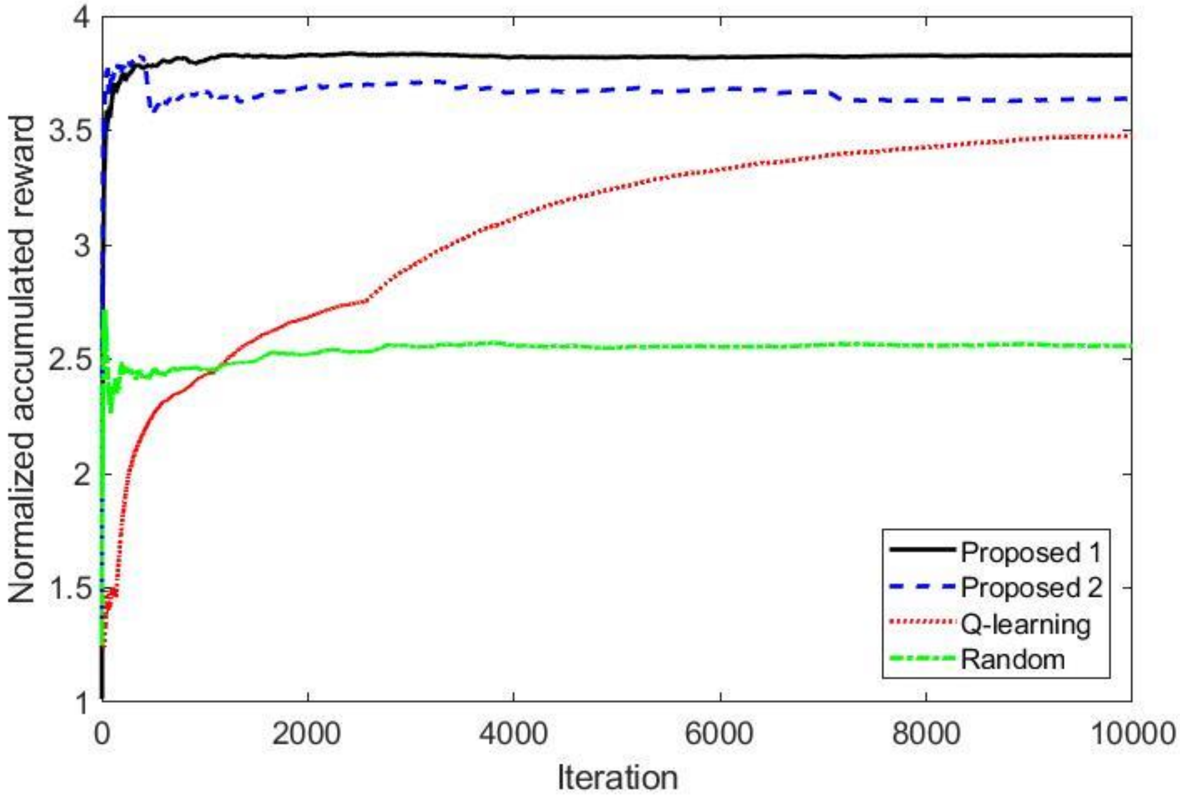




Simulation Results

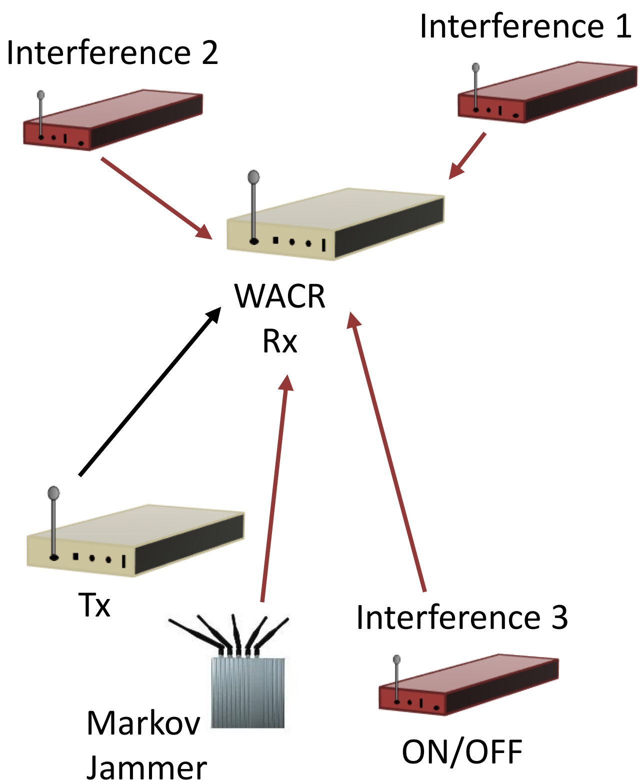
Test case 2:

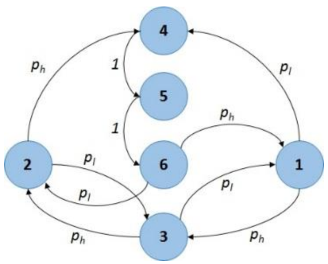


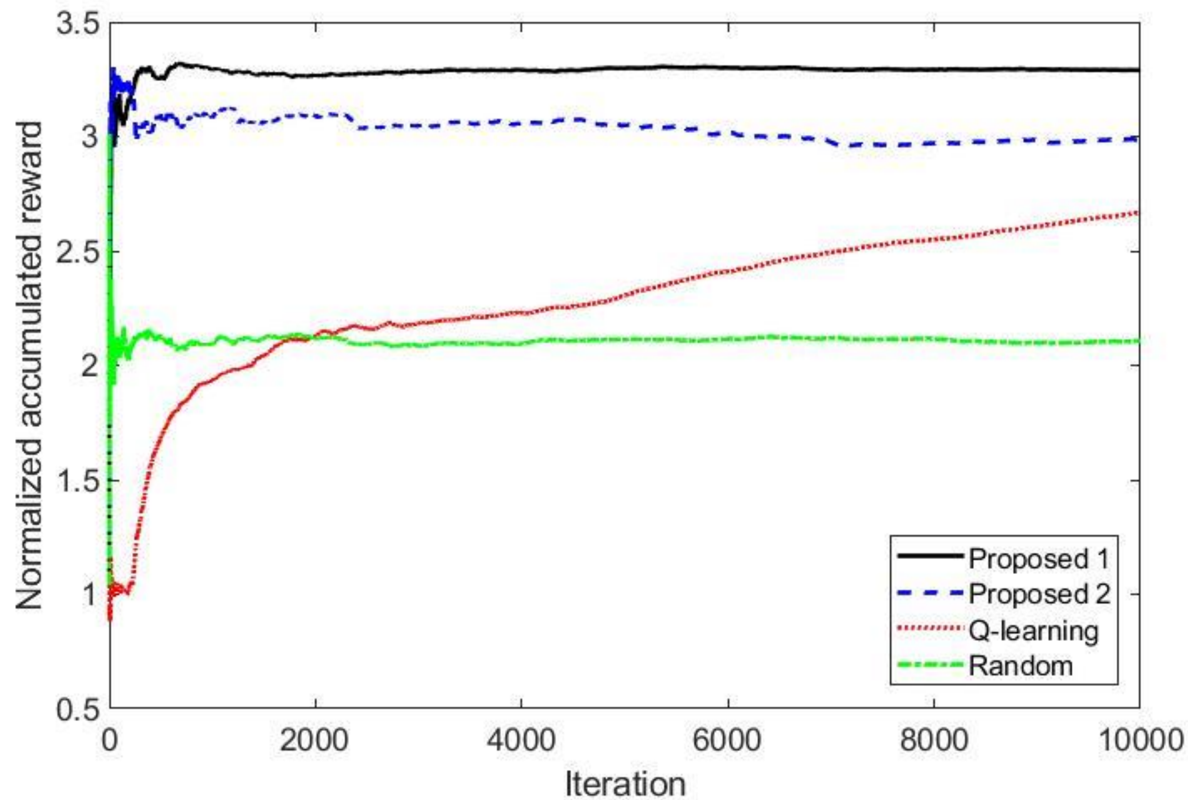


Simulation Results

Test case 3:







Simulation Results

Table I
PERFORMANCE COMPARISON: NORMALIZED ACCUMULATED REWARD VALUES AFTER 10,000 ITERATIONS.

Test case	Scenario	Proposed 1	Proposed 2	Q-learning	Random	Optimal
1	2 inter. signals	3.9	3.7	3.5	3.1	4
2	3 inter. signals	3.8	3.6	3.4	2.5	4
3	3 inter. signals and Markov jammer	3.3	3	2.7	2.1	4

Table II
CNN PARAMETERS OF THE PROPOSED ALGORITHMS.

	Input	Conv. 1	Conv. 2	Pool	FC	Comp.
Proposed 1	$3 \times 2 \times 3$	$2 \times 2 \times 10$	/	2×1	6	0.0196
Proposed 2	$4 \times 6 \times 1$	$1 \times 1 \times 10$	$2 \times 2 \times 20$	/	6	1

Questions

References

1. S. K. Jayaweera, "Signal processing for cognitive radios," John Wiley & Sons, 2015.

2. M. A. Aref, S. K. Jayaweera and S. Machuzak, "Multi-agent Reinforcement Learning Based Cognitive Anti-jamming", IEEE Wireless Communications and Networking Conference (WCNC'17), San Francisco, CA, Mar. 2017.

3. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, and G. Ostrovski, "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, Jan. 2015.

4. H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning,"The Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), Phoenix, AZ, USA, Feb. 2016.

5. M. A. Aref and S. K. Jayaweera, "Spectrum-agile cognitive interference avoidance through deep reinforcement learning," 14th EAI International Conference on Cognitive Radio Oriented Wireless Networks (CROWNCOM' 19), Poznan, Poland, Jun. 2019.