Reinforcement Learning for telecommunication networks: from Opportunistic Spectrum Access to IoTs

Raphaël Féraud

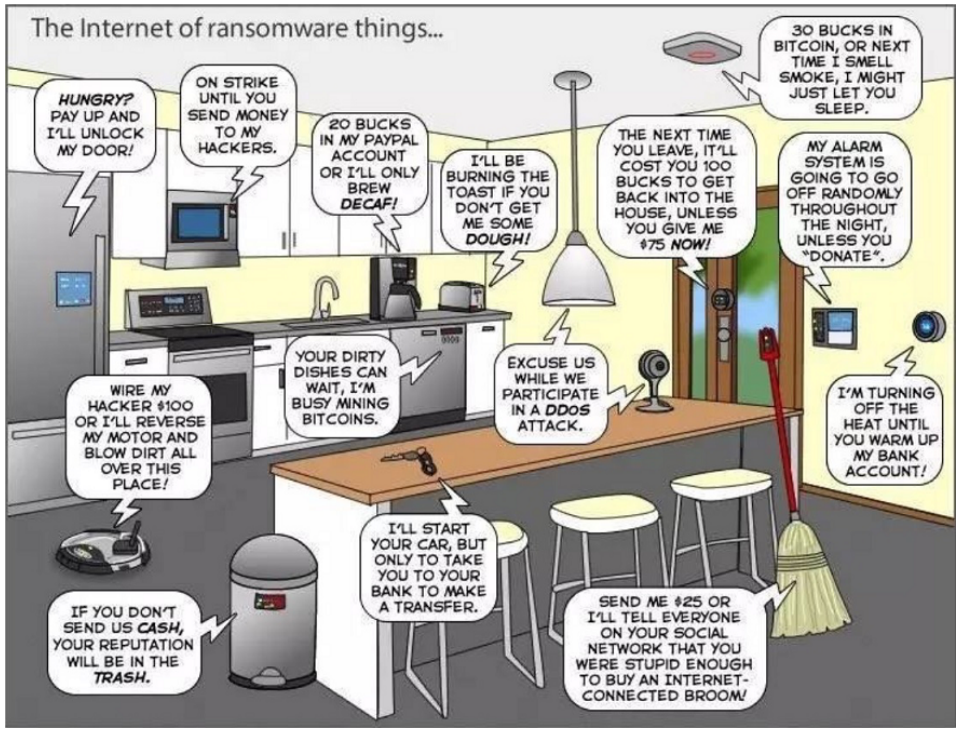
ORANGE Labs

June 2020

Outline

IoT and AI for a better world ?

IoT: Large system of connected objects that share data over the Internet.
AI: the study of how to produce machines that have some of the qualities that the human mind has.



Here we focus on the use of Reinforcement Learning for optimizing communications in IOT networks.

The Joy of Tech by Nitrozac and Snaggy

IoT networks: why optimizing communications ?

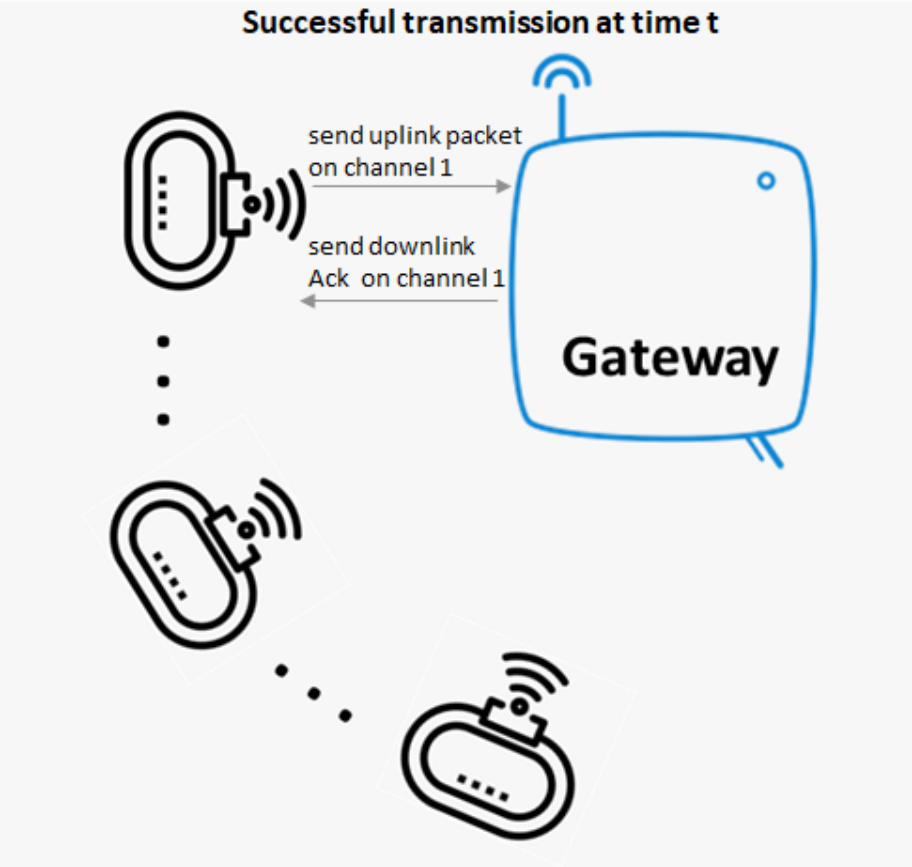
IoT networks: main characteristics

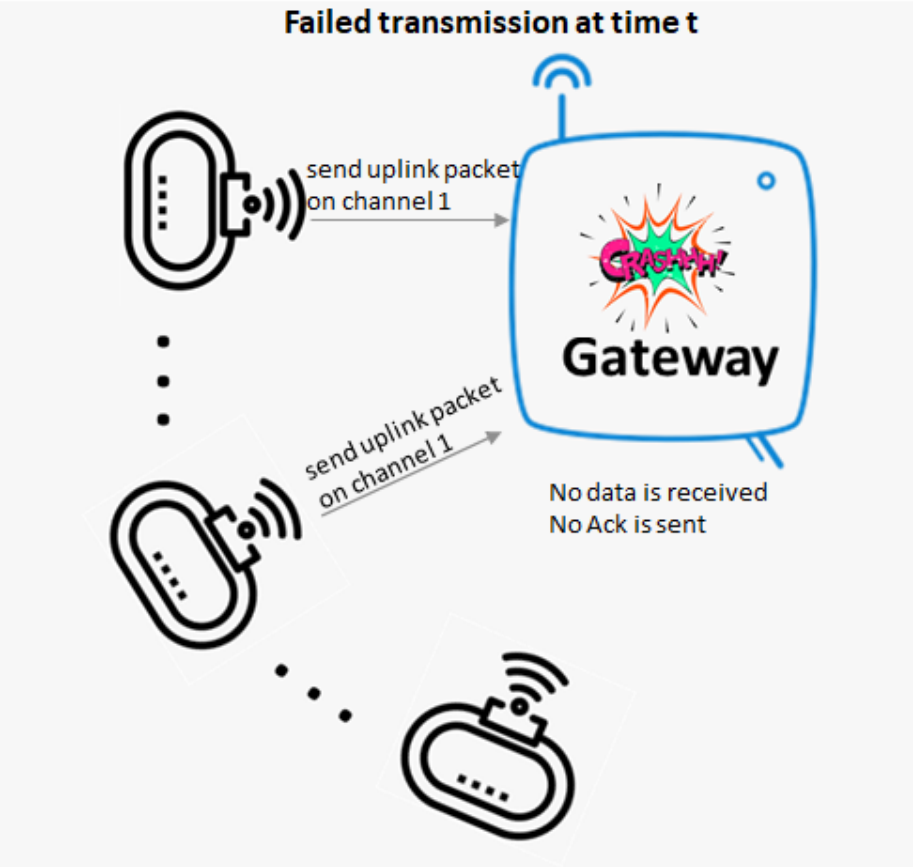
- Decentralized: devices initiate transmission
- Unlicensed radio bands
- Massive number of devices
- Low power devices
- Low duty cycle
- Low data rate

- High density of devices transmitting packets $\Longrightarrow$ **high congestion risk**.
- IoT networks use the repetition techniques to limit the **loss of messages** $\Longrightarrow$ **higher energy consumption** and **higher congestion risk**...

For limiting energy consumption we need to limit the loss of messages.

IoT networks: why some messages are lost ?



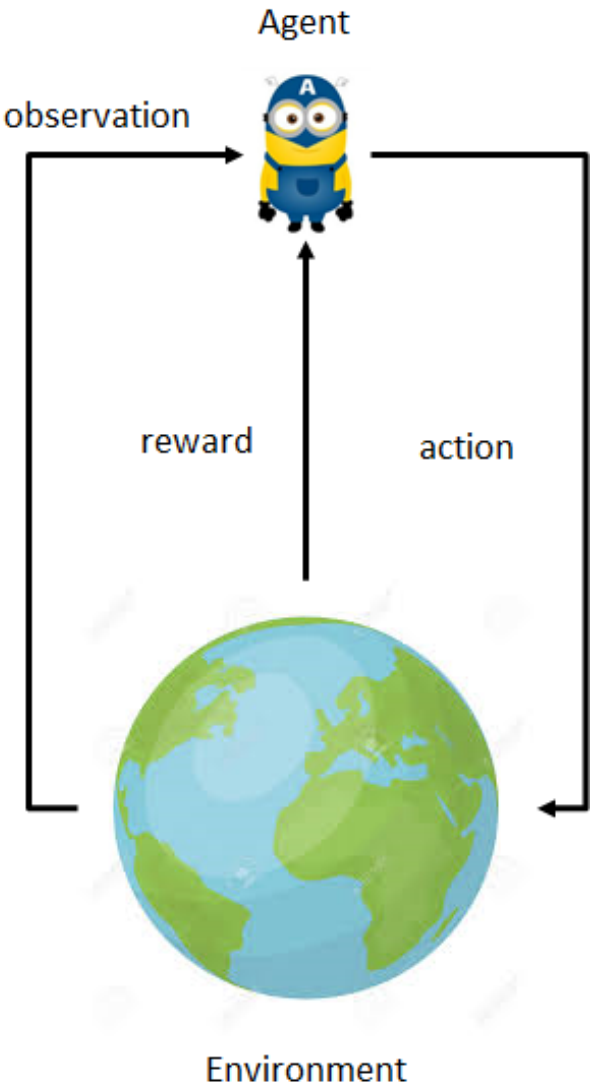


A collision occurs when two devices send an uplink packet at the same time on the same channel: the gateway does not received the packets and hence does not send acknowledges.

So how minimizing collisions ?

Outline

A generic problem: the reinforcement learning



The agent:

- Observes the environment,
- Executes an action,
- Receives a reward.

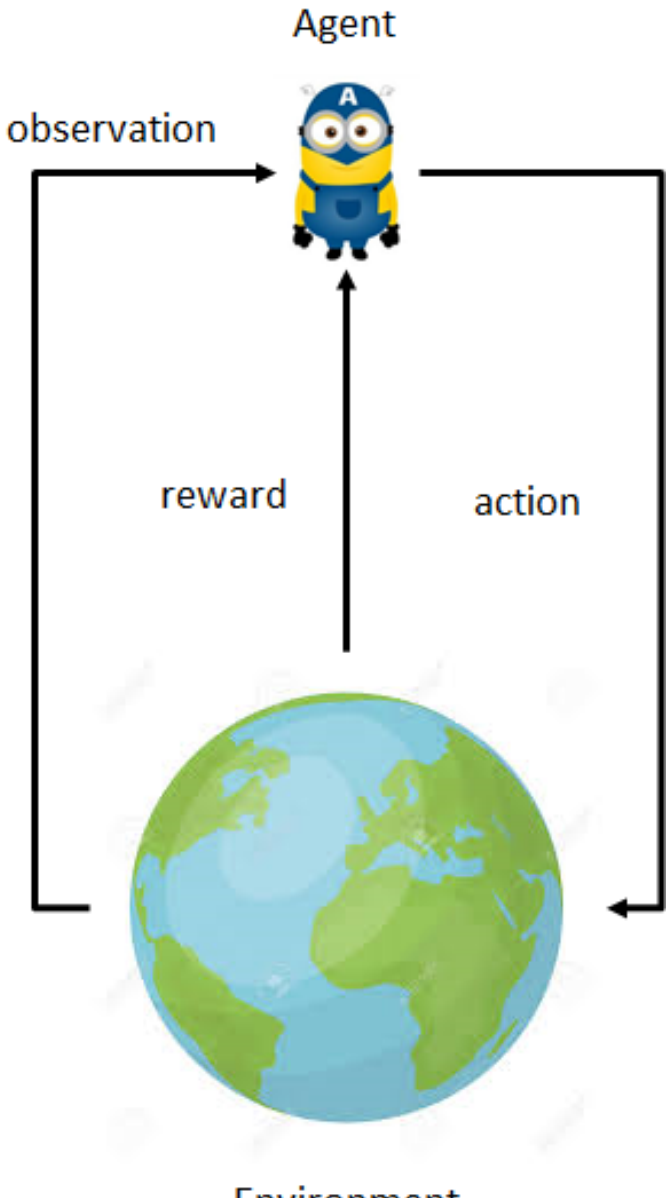
The environment:

- Changes its internal state due to the action,
- Emits a reward.

Goal:

Maximize the long term rewards of the agent.

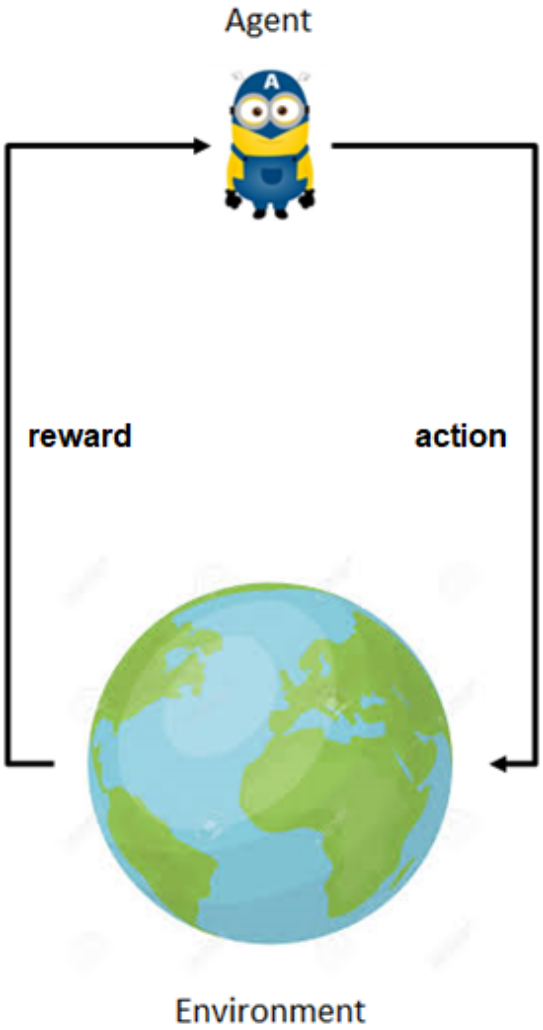
A generic problem: the reinforcement learning



Main features of reinforcement learning:

- There is no supervisor, only the reward of played action is revealed to the agent.
- The environment is initially unknown: the agent has to interact with the environment to gather information.
- Due to the change in the state of the environment, the actions of the agent affect the future rewards it will receive.
- The reward is delayed.

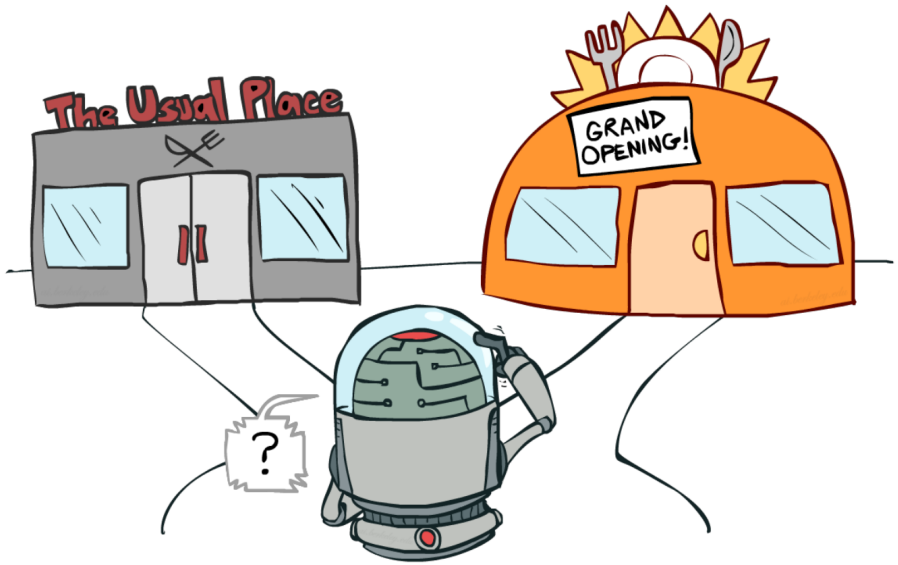
The multi-armed bandit problem



Main features of the multi-armed bandit problem:

- There is no supervisor, only the reward of played action.
- The environment is initially unknown: the agent has to interact with the environment to gather information.
- The played action does not affect the state of the environment.
- The rewards are not delayed.

Exploration / Exploitation dilemma



- Exploration: the agent plays a loosely estimated action in order to build a better estimate.
- Exploitation: the agent plays the best estimated action in order to maximize its cumulative reward.

Stochastic Bandits

Inputs: a set of arms $[K]$, unknown probability distributions of rewards $\upsilon_1, ..., \upsilon_K$, and unknown mean rewards $\mu_1, ..., \mu_k$

1: **for** $t = 1, 2, ...,$ **do**
2: Player chooses $k_t \in [K]$
3: Environment reveals $r_{k_t} \sim \upsilon_{k_t}$
4: **end for**

The goal of the player is to minimize the pseudo-regret with respect to the optimal policy:

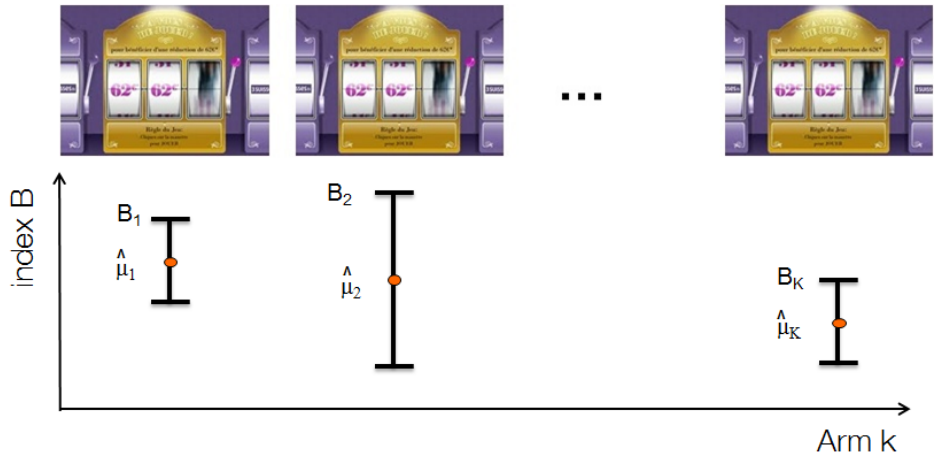
$$R(T) = \max_{k \in [K]} \mu_k . T - \mathbb{E}_\upsilon \sum_{t=1}^{T} r_{k_t} = \mu_{k^*} . T - \mathbb{E}_\upsilon \sum_{t=1}^{T} r_{k_t}.$$

Any algorithm has a pseudo-regret at least to (Lai and Robbins 1985):

$$\liminf_{T \to \infty} R(T) \geq \sum_{k=1}^{K} \frac{\Delta_k}{KL(\upsilon_k, \upsilon_{k^*})} \log T = \Omega\left(\frac{K}{\Delta} \log T\right),$$

where KL denotes the Kullback-Leiber divergence, $\Delta_k = \mu_{k^*} - \mu_k$, $\Delta = \mu_{k^*} - \max_{k \neq k^*} \mu_k$.

The optimism in face of uncertainly: Upper Confidence Bound

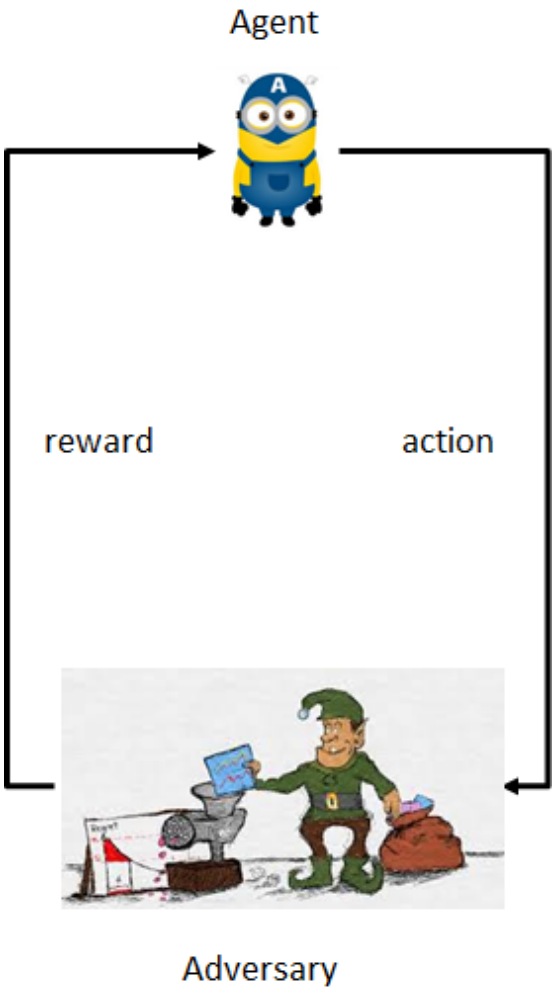


1: $\forall k, B_k := \infty$
2: **for** $t = 1, 2, ...,$ **do**
3: Player chooses: $k_t := \arg\max_{k \in [K]} \underbrace{\hat{\mu}_k}_{\text{estimated mean reward}} + \underbrace{\sqrt{\frac{2 \log t}{t_k}}}_{\text{confidence interval } B_k}$
4: Environment reveals $r_{k_t} \sim \upsilon_{k_t}$
5: $t_{k_t} := t_{k_t} + 1$, player updates $\hat{\mu}_{k_t}$
6: **end for**

Regret Upper Bound (Auer et al 2002):

$$R(T) \leq O\left(\frac{K}{\Delta} \log T\right)$$

Adversarial Bandits



Main feature of adversarial bandits:

- An adversary, which knows the algorithm of the player, has generated a deterministic sequence of rewards for each action:
  - the player has to randomize the choice of arms,
  - the player has to continuously explore overtime.
- The time horizon is known.
- The player competes against the best arm of the run: $\max_{k\in[K]} \sum_{t=1}^{T} r_k(t)$

Lower Bound (Auer et al 2001):

$$\Omega\left(\sqrt{KT}\right)$$

Exp3: Exponential-weight algorithm for Exploration and Exploitation

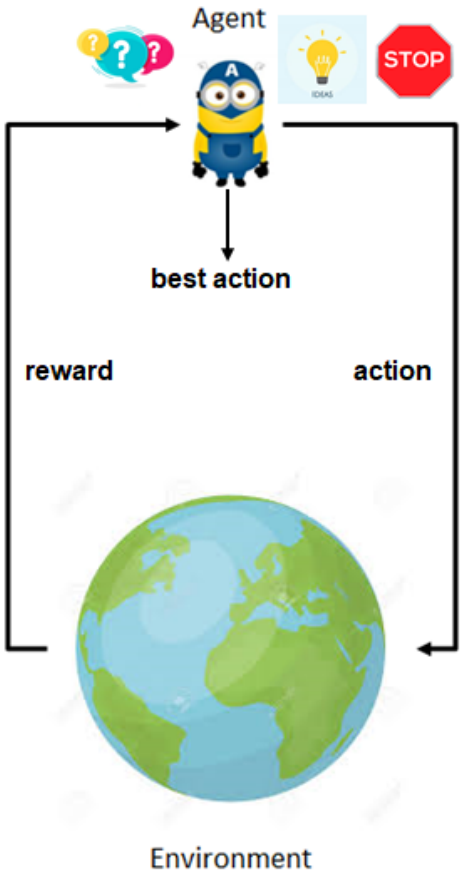
Input: $\gamma \in (0,1]$

1: $\forall k,\ w_k(1) := 1,\ t := 1$
2: **for** $t = 1, 2, ...,$ **do**
3: $\quad \forall k$: $p_k := (1-\gamma)\frac{w_k(t)}{\sum_k w_k(t)} + \frac{\gamma}{K}$ *(constant exploration rate)*
4: $\quad$ Player chooses: $k_t \sim (p_1, ..., p_K)$
5: $\quad$ Adversary reveals r_{k_t}
6: $\quad \hat{r}_{k_t}(t) := \frac{r_{k_t}(t)}{p_{k_t}(t)}$ *(unbiaised estimation of $r_{k_t}(t)$)*
7: $\quad w_{k_t}(t+1) := w_{k_t}(t) \exp\left(\gamma \hat{r}_{k_t}(t)/K\right)$
8: $\quad t := t+1$
9: **end for**

Regret Upper Bound (Auer et al 2001):

$$\text{Choosing } \gamma := \min\left(1, \sqrt{\frac{K \log K}{(e-1)T}}\right), \text{ we have: } R(T) \leq O\left(\sqrt{KT \log K}\right)$$

Best Arm Identification problem



Inputs: a set of arms $[K]$, unknown probability distributions of rewards $\upsilon_1, ..., \upsilon_K$, and unknown mean rewards $\mu_1, ..., \mu_k$, $\varepsilon \geq 0$, $\delta \in (0,1)$, $t := 1$
Output: one arm

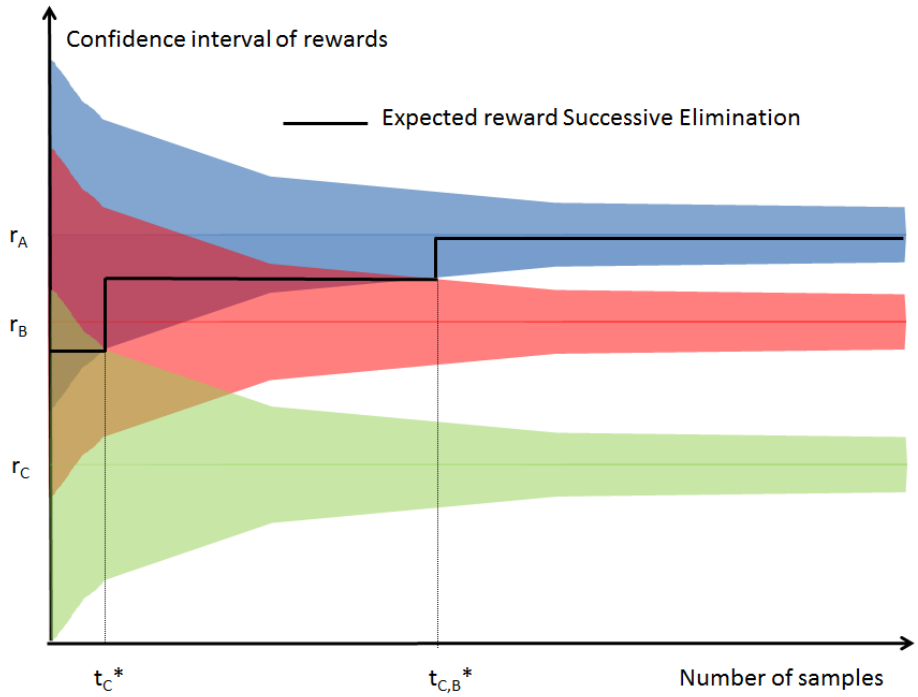
1: **repeat**
2: Player chooses $k_t \in [K]$
3: Environment reveals $r_{k_t} \sim \upsilon_{k_t}$
4: $k' = \arg\max_{k \in [K]} \hat{\mu}_k$
5: $t := t + 1$
6: **until** $\mathbb{P}\{\mu_{k^*} - \mu_{k'} > \varepsilon\} \leq \delta$

Lower Bound (Mannor and Tsitsiklis 2004):

The sample complexity, i.e. the stopping time, of any algorithm is at least:

$$\Omega\left(\frac{K}{\Delta^2} \log \frac{1}{\delta}\right)$$

Successive Elimination



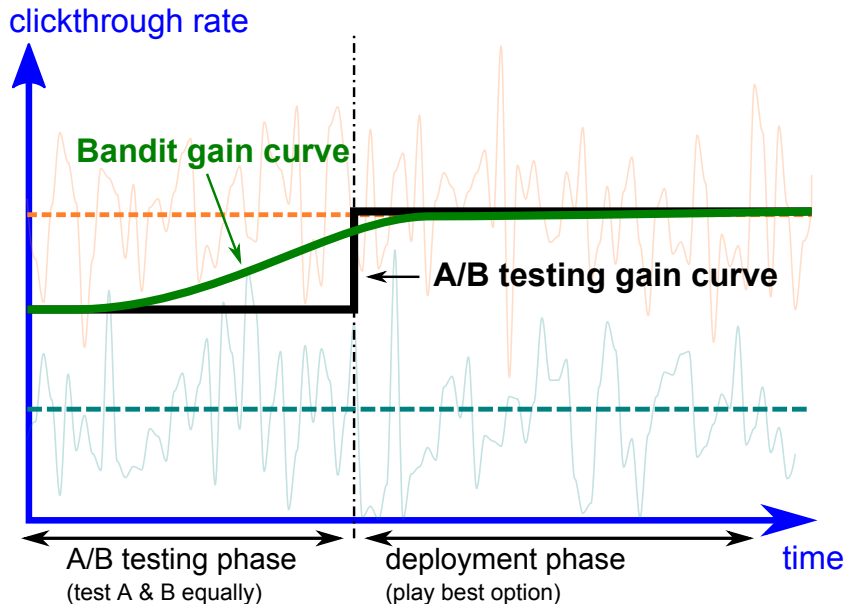
Inputs: a set of arms $[K]$, $\delta \in (0,1)$, an $\varepsilon \geq 0$
Output: the best arm

1: **repeat**
2: a remaining arm is uniformly sampled
3: Environment reveals $r_k \sim \upsilon_k$
4: $k' = \arg\max_{k \in [K]} \hat{\mu}_k$
5: $\forall k \in [K]$, **if** $\mu_{k'} - \mu_k + \varepsilon \geq 2\sqrt{\frac{1}{2t_k} \log \frac{4Kt_k^2}{\delta}}$
then remove k from $[K]$
6: **until** $|[K]| = 1$

Sample Complexity Upper Bound (Even-Dar et al 2006):

$$O\left(\frac{K}{\Delta^2} \log \frac{K}{\delta}\right)$$

Best Arm Identification versus Multi-Armed Bandits



The expected gain of Multi-Armed Bandits algorithms is higher than the one of Best Arm Identification algorithms.

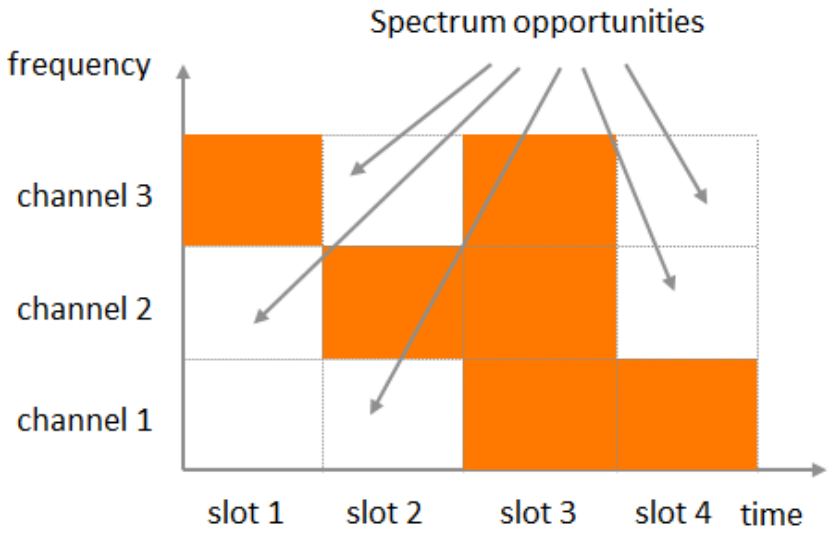
Best Arm Identification algorithms are used when there is a maintenance cost associated with each action: continuous development, clinical trials, phone marketing campaigns, blacklisting channels...

Some references

- Reinforcement Learning, Second Edition An Introduction, Richard S. Sutton and Andrew G. Barto, 2018
- Reinforcement Learning (RL) Course - YouTube: A 10-lecture course by David Silver.
- Asymptotically efficient adaptive allocation rules, Lai T. L., and Robbins H., Advances in Applied Mathematics, 1985.
- Finite-time Analysis of the Multiarmed Bandit Problem, Auer, P., Cesa-Bianchi, N. and Fischer, P., Machine Learning 47, 2002.
- The Nonstochastic Multiarmed Bandit Problem, Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire, SIAM J. Comput., 32(1), 2001.
- The Sample Complexity of Exploration in the Multi-Armed Bandit Problem, Mannor S., and Tsitsiklis J. N., Journal of Machine Learning Research, 2004.
- Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems, Even-Dar, E., Mannor, S., Mansour Y., JMLR, 2006.

Outline

Opportunistic Spectrum Access



OSA networks: Principle

- Primary Users are on licensed channels.
- Secondary Users have the opportunities to send data on free slots:
  - Secondary Users first do sensing on one channel, and try to send a packet if there is no Primary Users,
  - if two or more Secondary Users send data on the same channel a collision occurs.

Multi-Player Multi-Armed Bandits

- K arms (channels) with different and unknown vacancy rates $\theta^k \in [0,1]$
- $N \leq K$ players (Secondary Users)
- synchronized players: at each time slot each player sends a packet.

At each time slot $t \in 1, ..., T$

- player n selects arm k_n
- player does sensing on arm k_n: observe $Y^{k_n}(t) \sim B(\theta^{k_n})$
- if $Y^{k_n}(t) = 1$ (channel k_n is free) player n sends packet and observes $C^{k_n}(t) = 1$ if another player chooses the same arm, and $C^{k_n}(t) = 0$ else.

Goal

- maximize the sum of rewards of all players: $\sum_{n=1}^{N} \sum_{t=1}^{T} Y^{k_n}(t).(1 - C^{k_n}(t))$
- trade-off exploration / exploitation / collisions.

Regret for Multi-Player Bandits

Lower bound (Besson and Kaufmann 2018)

Assume that $\mu_1 \leq \mu_2 \leq ... \leq \mu_K$, then we have for any algorithm:

$$\liminf_{t\to\infty} R(T) \geq N \sum_{k=N+1}^{K} \frac{\mu_k - \mu_N}{KL(\mu_k, \mu_N)} \log T = \Omega\left(\frac{N(K-N+1)}{\mu_N - \mu_{N+1}} \log T\right)$$

Regret Decomposition (Besson and Kaufmann 2018:)

Let $T^k(t)$ be the number of plays of arm k at time t. It exists $A, B \in (\mathbb{R}^+)^2$, such that:

$$R(T) \leq A \underbrace{\sum_{k=N+1}^{K} \mathbb{E}[T^k(T)]}_{\text{number of plays of sub-optimal arms}} + B \underbrace{\sum_{k=1}^{K} \mathbb{E}[C^k(T)]}_{\text{number of collisions}}$$

First idea:

- Find the N-best arms,
- Use an orthogonalization procedure to avoid collisions.

Musical Chairs (Rosenski et al 2018)

Inputs: a set of arms $[K]$, approximation factor $\varepsilon \in (0,1]$, probability of failure $\delta \in (0,1)$

$T_0 := \lceil \frac{16K}{\varepsilon^2} \log \frac{4K^2}{\delta} \rceil$
for $t = 1,2,...,T_0$ **do** *(Estimate the values of arms and the number of collisions)*
    player n plays $k_n \sim U(1,...,K)$
    Environment reveals $Y^{k_n}(t) \sim B(\theta_{k_n})$ and $C^{k_n}(t)$
    Evaluate $\hat{\mu}^{k_n}(t)$, and the number of collisions $C_n(t)$
end for
$s_n := 0$, $\hat{N}_n := \lfloor \frac{\log \frac{T_0 - C_n(T_0)}{T_0}}{\log(1-1/K)} + 1 \rfloor$ *(Estimate the number of players)*
Sort arms in decreasing order of $\hat{\mu}_k$
for $T_0, T_0+1, ...$ **do** *(Musical Chairs)*
    if $s_n = 0$ **then**
        Play $k_n \sim U(1,...,\hat{N}_n)$ *(Play a $\hat{N}_n$-best arms)*
        if $Y^{k_n}(t) = 1$ $C^{k_n}(t) = 0$ **then** $s_n := 1$
    else Play k_n *(Stays on arm k_n)*
    end if
end for

Analysis:

$$R(T) \leq O\left(\frac{NK}{\varepsilon^2} \log T\right)$$

MC-TOP-N algorithm (Besson and Kaufmann 2018)

Idea: at time t stay on the same arm ($s_n = 1$) if it is good ($k_n \in \mathcal{N}_n(t)$) and free ($C^{k_n}(t) = 0$).

Inputs: a set of arms $[K]$, N players, $\forall n$, $k_n \sim \mathcal{U}(1,...,K)$, $s_n := 0$

1: **for** $t = 1, 2, ...$ **do**
2: player n chooses $k_n = \arg\max_{k \in [K]} UCB_n^k(t)$
3: Environment reveals $Y^{k_n}(t) \sim B(\theta^{k_n})$
4: Compute $UCB_n^k(t)$
5: $\mathcal{N}_n(t) := \{$arms with the N largest $UCB_n^k(t)\}$
6: **if** $k_n \notin \mathcal{N}_n(t)$ **then** player n chooses k_n uniformly in $\mathcal{N}_n(t)$ and $s_n := 0$
7: **elseif** $C^{k_n}(t) = 1$ and $s_n = 0$ **then** player n chooses k_n uniformly in $\mathcal{N}_n(t)$ and $s_n := 0$
8: **else** $s_n := 1$ *(player n is fixed on arm k_n)*
9: **end for**

Analysis:

$$R(T) \leq O\left(\frac{N(K-N+1)}{\mu_N - \mu_{N+1}} \log T\right)$$

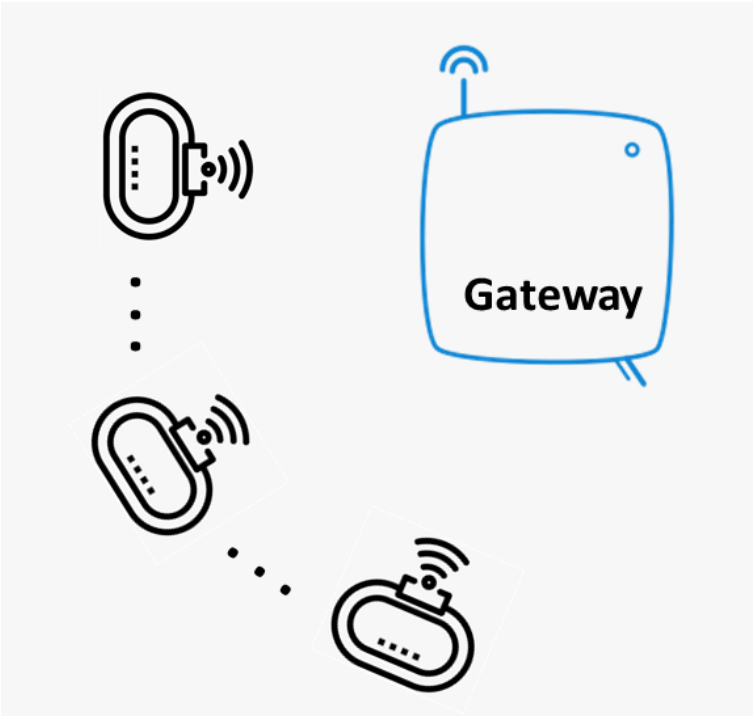
Some references

- Distributed learning in Multi-Armed Bandit with multiple players, K. Liu and Q. Zhao, IEEE Transaction on Signal Processing, 2010.
- Distributed algorithms for learning and cognitive medium access with logarithmic regret, A. Anandkumar, N. Michael, A. K. Tang, and S. Agrawal, IEEE Journal on Selected Areas in Communications, 2011.
- Multi-player bandits: a musical chairs approach, Jonathan Rosenski, Ohad Shamir, and Liran Szlak, ICML 2016.
- Multi-Player Bandits Revisited, Lilian Besson, Emilie Kaufmann, ALT 2018.
- Distributed Multi-Player Bandits - a Game of Thrones Approach, Ilai Bistritz, Amir Leshem, NeurIPS 2018.
- Selfish Robustness and Equilibria in Multi-Player Bandits, E. Boursier, V. Perchet, COLT 2020.

Outline

Problem Formulation

- **N Asynchronous Players:** at each time slot each player n has a probability p_n to be active (i.e. to send data to the gateway)
- **K Arms** are available to all players, $N >> K$
- A transmission is successful when it does not collide:
  - **External collision:** $E^k \sim B(\theta^k)$ (equals 0 if collision, 1 otherwise)
  - **Internal collision:** I^k (equals 0 if collision, 1 otherwise) between controlled players
- Only the binary outcome is observed when playing arm k: $Y^k = E^k I^k$



Main differences in comparison to OSA:

- large number of asynchronous players ($N >> K$),
- No sensing: collisions are not observed

Why it is a very challenging problem ?

Policy: How the players play the arms

- $\pi = (\pi_1, ..., \pi_N)$
- $\pi_n = (\pi_n^1, ..., \pi_n^K)$: policy of player n
- π_n^k: probability that player n chooses arm k when active

Expected Reward per time slot

$$\mu(\pi) = \sum_{k=1}^{K} \underbrace{\theta^k}_{\texttt{mean reward of arm k}} \sum_{n=1}^{N} \underbrace{p_n.\pi_n^k}_{\texttt{player n chooses arm k}} \underbrace{\prod_{n'=1, n' \neq n}^{N} (1 - p_{n'}.\pi_{n'}^k)}_{\texttt{no collision occurs}}$$

The optimization problem is not convex $\Longrightarrow$ efficient optimization methods cannot be applied to it.

Open problem: Is it NP-Hard ?

The selfish heuristic

Each device selfishly optimizes its choice of arms:

- $a_n \sim B(p_n)$
- if $a_n = 1$ (device is active) then
  - player n chooses arm $k_n \in [K]$ using a MAB algorithm
  - player n observes the outcome of arm k_n: $Y^{k_n} = E^{k_n} I^{k_n}$
  - player n updates the MAB algorithm

This is an heuristic:

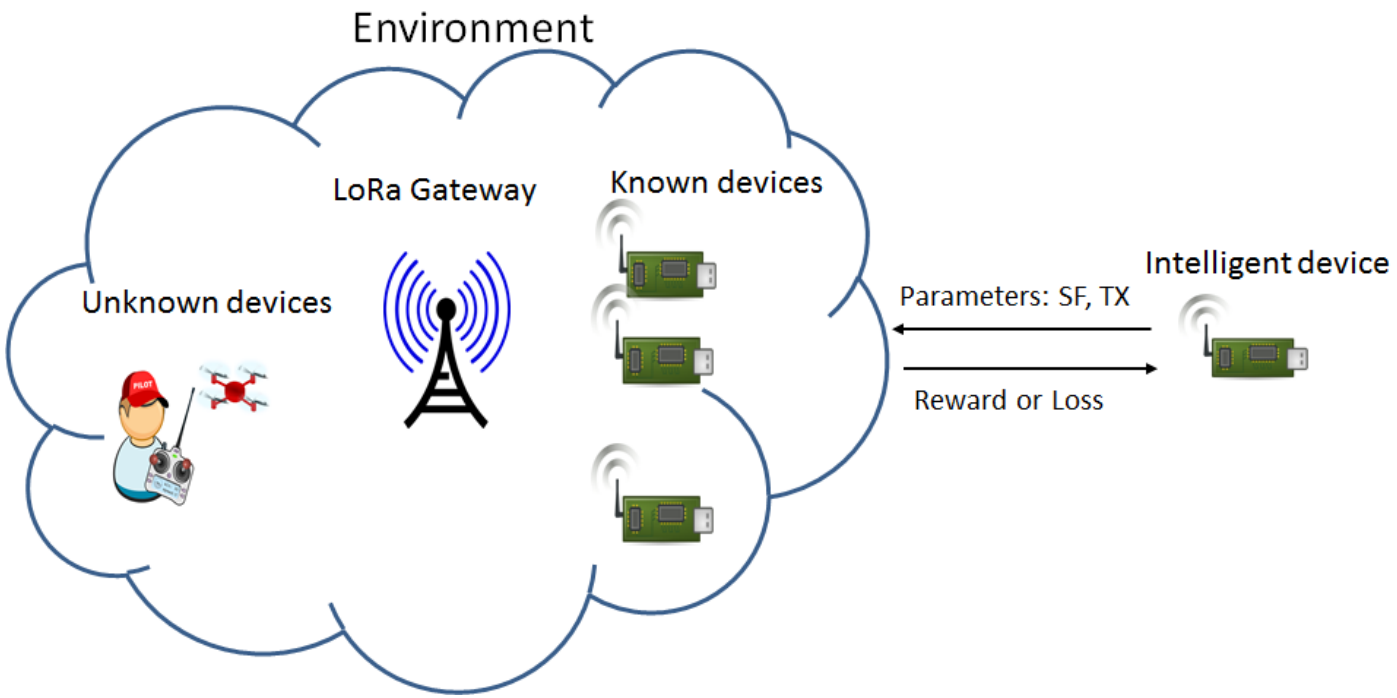
The above algorithm run on each player does not reach the optimum of the optimization problem, and there is no guaranty to reach an equilibrium.

From the player point of view, this is not a stochastic MAB problem:

due to the collisions with other learning players the rewards of arms change during time.

But it works well !

Optimization of LoRa transmissions



The device chooses the parameters for the next LoRa transmission, then the environment, which is the gateways, the other devices, the weather..., generates a reward that is used by the device to optimize the choice of parameters.

Experimental setting

Simulator:

A realistic simulator, which implements propagation models, shadowing, fast fading, collision rules and retransmissions (N. Varsier and J. Schwoerer 2017), is used for the experiments.

One versus all:

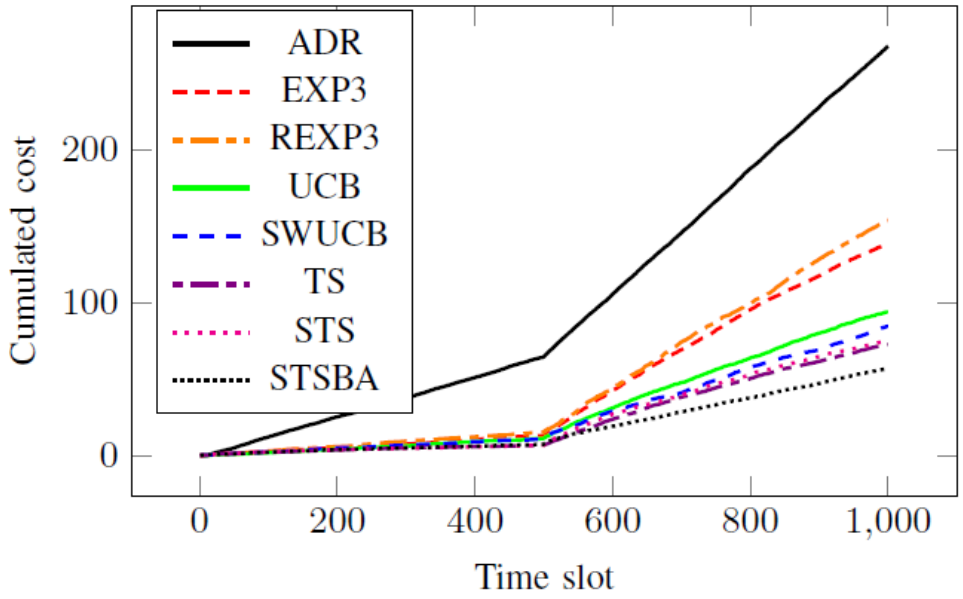
A single indoor gateway with 100 nodes is considered:

- 99 nodes, that are randomly positioned within a radius of 2 km, use the algorithm defined in the LoRa protocol: Adaptive Data Rate (ADR).
- 1 node uses bandit algorithms for choosing transmission parameters.

Scenario:

1. 1000 packets are sent by the optimized node. Due to the use of ADR on 99 nodes, the mean reward of each arm evolves during time.
2. After sending 500 packets, the node moves from 592 meters to 1975 meters from the gateway. The mean reward of arms abruptly changes.

Results



Comparison with ADR:

ADR algorithm is clearly dominated by any MAB algorithm.

Multi-Armed Bandits:

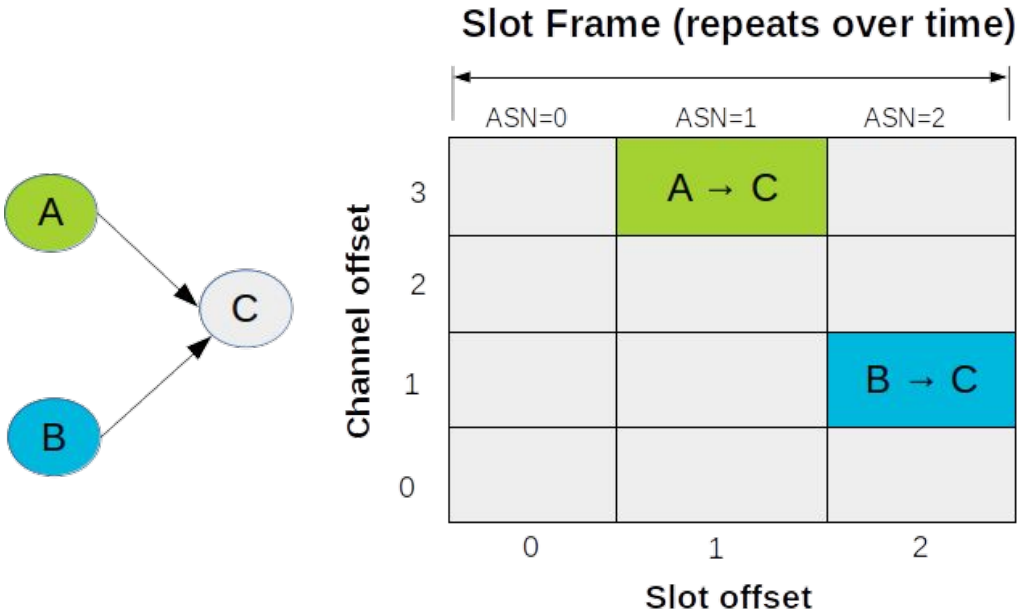
Switching Thompson Sampling with Bayesian Aggregation is the best-performing algorithm. Surprisingly Thompson Sampling performs as well as STS and SWUCB, which are designed for switching environments.

Some references

- Capacity limits of LoRaWAN technology for smart metering, N. Varsier and J. Schwoerer, IEEE International Conference on Communications, 2017
- Multi-armed bandit learning in IoT networks; learning helps even in non-stationary settings, R. Bonnefoi, L. Besson, C. Moy, E. Kaufmann, and J. Palicot, CROWN, 2018
- Multi-player bandits revisited, L. Besson, and E. Kaufmann, ALT 2018.
- Node-based optimization of LoRa transmissions with multi-armed bandit algorithms, R. Kerkouche, R. Alami, R. Féraud, N. Varsier, and P. Maillé, ICT 2018.
- Memory Bandits: a Bayesian approach for the Switching Bandit Problem R. Alami, O. A. Maillard, R. Féraud, NIPS Workshop on Bayesian Optimization, 2017.
- Restarted Bayesian Online Change-point Detector achieves Optimal Detection Delay, R. Alami, O. A. Maillard, R. Féraud, ICML 2020.

Outline

IEEE 802.15.4 TSCH (Time Slotted Channel Hoppings) network



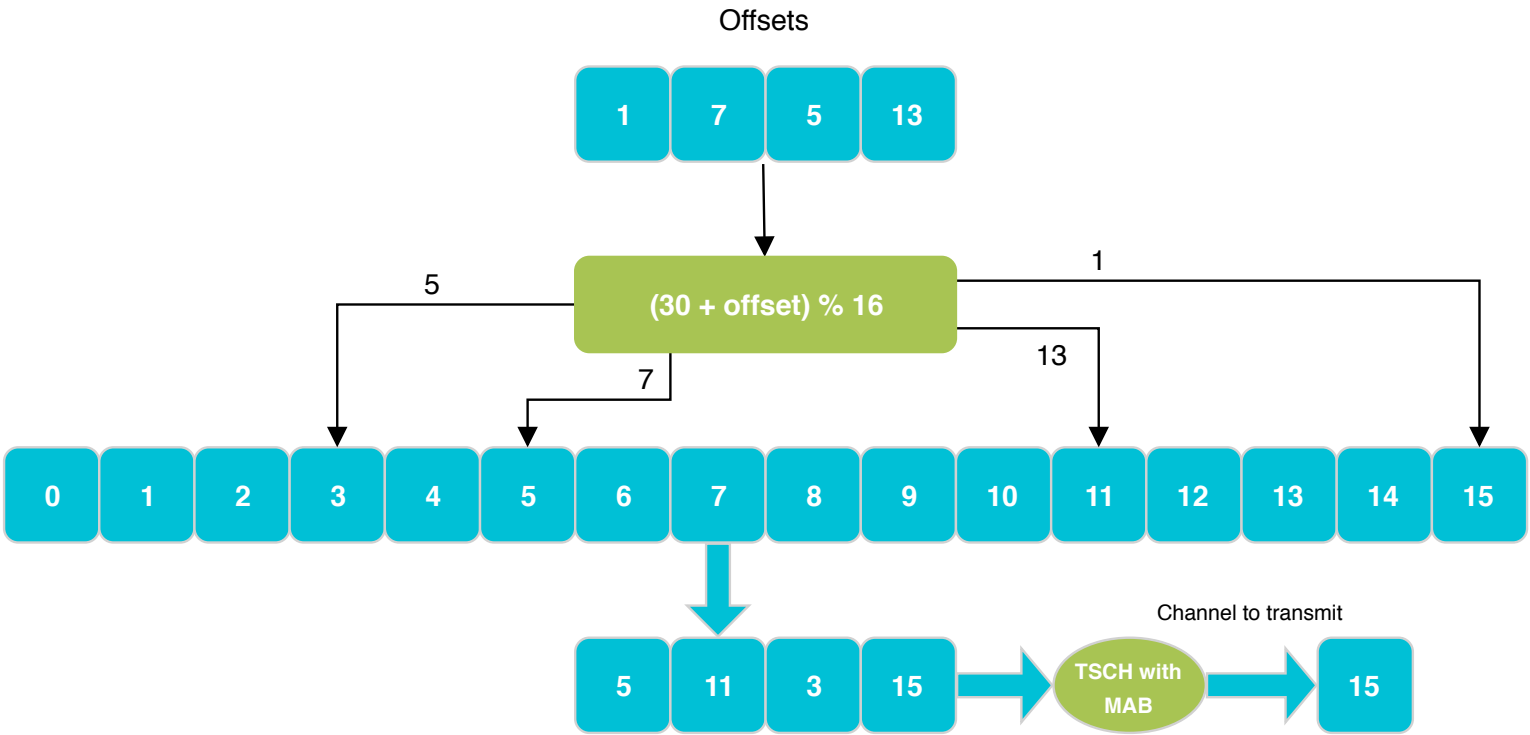
- Communications are coordinated by a scheduler (time slot x set of channels): avoids internal collisions.
- Interference issues on the license free bandwidths: external collisions occur.
- To prevent a node n from being systematically assigned to a bad channel $k \in [K]$, TSCH implements a channel hopping function by assigning each node to a different offset:

$$k_n(t) = (t + Offset_n) \mod K$$

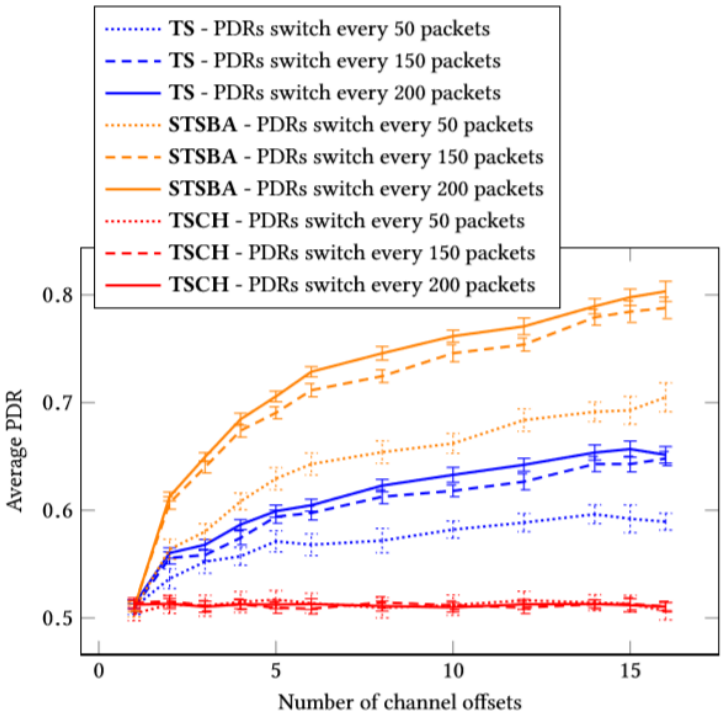
Multi-armed Bandit Algorithms for TSCH-compliant channel selection

The setting

- When a node sends a packet, it receives an ack: the reward is the success of the transmission.
- each node is assigned to a subset of offsets.
- a MAB algorithm is used to select the best offset in its subset.



Performance gain



- Varying the size of the subset of offset from 1 to 16.
- Packet Data Rate of channels switches every 200 packets.
- Comparing the performances between Thompson Sampling (TS), Switching Thompson Sampling with Bayesian Aggregation (STSBA), and standard TSCH.

Best algorithm: Switching Thompson Sampling with Bayesian Aggregation.
The performance of TSCH without MAB does not depend on the size of subset since the choice of the offset is random.

Reference: Reinforcement Learning techniques for optimized channel hopping in IEEE 802.15.4 TSCH networks, H. Dakdouk, E. Tarazona, R. Alami, R. Féraud, G. Z. Papadopoulos, and P. Maillé, ACM MSWIM, 2018.

Outline

Conclusion

- Multi-Armed Bandit handles a central problem in reinforcement learning: the exploration / exploitation dilemma.
- Multi-Armed Bandit algorithms are efficient (low consumption of resources) and are equipped with theoretical guarantees.

- Opportunistic Spectrum Access can be formulated as a multi-player multi-armed bandits: optimal solutions exist.
- Channel allocation in IoT networks is a more challenging problem (asynchronous players, large number of players, collisions not observed):
  - the selfish UCB heuristic works well in practice,
  - providing algorithms equipped with theoretical guarantee is still an open problem.
- many other problems in telecommunication networks can be handled using Multi-Armed Bandits: auto-configuration in SON network, sensor networks...