

1. INTRODUCTION & GOAL

Goal: fit more objects in a "Internet of Things" networks, keep a good *Quality of Service*.

- *Hypothesis*: objects choose channel $k \in \{1, \dots, K\}$, to use for each communication.
- *Idea*: use on-line **Machine Learning algorithms** ?
- *Not so easy*: each device takes its own decisions, without central control or communication, has light CPU/memory etc. . .
- \Rightarrow **Solution: Decentralized MAB algorithms !**

3. SOME BASELINE ALGORITHMS

Performance = *successful transmission rate*.
Three algorithms used for baseline comparison.

- **Naive algorithm**: all the D dynamic devices choose their channel $k_i(t) \sim U(\{1, \dots, K\})$ purely uniformly at random.
- **Optimal algorithms**: exact algorithm (or a greedy approximation), when a *centralized agent* can affect the D dynamic devices to channels.



Inapplicable in practice as we need a decentralized approach, but it gives a *baseline* for comparison.

4. MULTI-ARMED BANDITS ALGORITHMS

Every time $t \in \mathbb{N}^*$ a dynamic device needs to send :

1. it **chooses a channel** $A(t) \in \{1, \dots, K\}$
2. it sends an **uplink packet** \nearrow on that channel
2. then it **observes a binary reward** $r_A(t) \in \{0, 1\}$ (1 if *Ack* \checkmark is well received, 0 if collision)

4.1. UPPER CONFIDENCE BOUND ALGO.

Simple *frequentist* approach :

- Selections of channel k , up-to time t
$$N_k(t) := \sum_{\tau=1}^t \mathbb{1}(A(\tau) = k)$$
- Accumulated rewards
$$X_k(t) := \sum_{\tau=1}^t r_k(\tau) \times \mathbb{1}(A(\tau) = k)$$
- UCB₁ uses a *confidence term* (parameter $\alpha > 0$)
$$B_k(t) := \sqrt{\alpha \log(t) / N_k(t)}$$
- To compute its *index* (upper confidence bound)
$$U_k(t) := X_k(t) / N_k(t) + B_k(t) = \widehat{\mu}_k(t) + B_k(t)$$
- Use $U_k(t)$ to decide the channel for next step:
$$A(t+1) \in \arg \max_{1 \leq k \leq N_c} U_k(t)$$

\Rightarrow UCB₁ is a *deterministic index policy*.

4.2. THOMPSON SAMPLING ALGORITHM

Old algorithm (1935), *Bayesian* approach :

- Start with a flat Beta prior, Beta(1, 1), on the (unknown) parameter $\mu_k \in [0, 1]$
- And at time t , the posterior counts the *successes* and *failures* of channel k :
$$\Pi_k(t) = \text{Beta}(1 + X_k(t), 1 + N_k(t) - X_k(t))$$
- Then *sample* a random *index* for each channel, from the posteriors:
$$I_k(t) \sim \Pi_k(t)$$
- And choose:
$$A(t+1) \in \arg \max_{1 \leq k \leq N_c} I_k(t)$$

\Rightarrow TS is a *randomized index policy*.

2. MODEL: TIME/FREQUENCY PROTOCOL DEVICES IN THE NETWORK

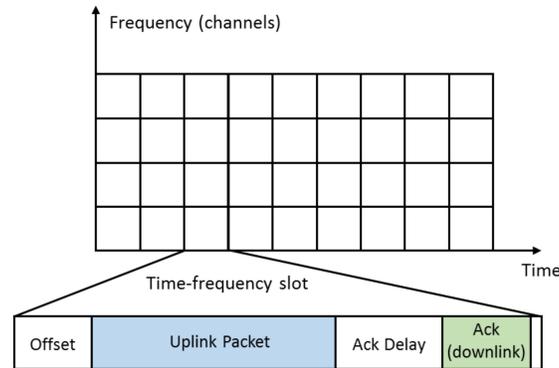


Figure 1: Time-frequency slotted protocol.

Frame = fix-duration *uplink slot* \nearrow (end-devices transmit their packets) + *Ack delay* + *downlink slot* \checkmark (base station replies with *Ack* if packet well received).

Model: One base station 

$K = 10$ RF channels (of same bandwidth).

S  + D  = 2000 end-devices in the network, with *very low duty-cycle* (one message every 1000 frame).

They are separated into *two groups*:

- S **static devices**  : poor RF abilities, and use only one channel to communicate with the base station. Their choice is fixed in time (stationary) and independent (*i.i.d.*). **interfering traffic** generated by static devices. (Unknown) affection to the K channels: $S = (S_1, \dots, S_K)$.
- D **dynamic devices**  : richer RF abilities, can use all the available channels, by quickly *reconfiguring their RF transceiver on the fly* (dynamically).

5. QUICK CONVERGENCE OF MAB ALGORITHMS

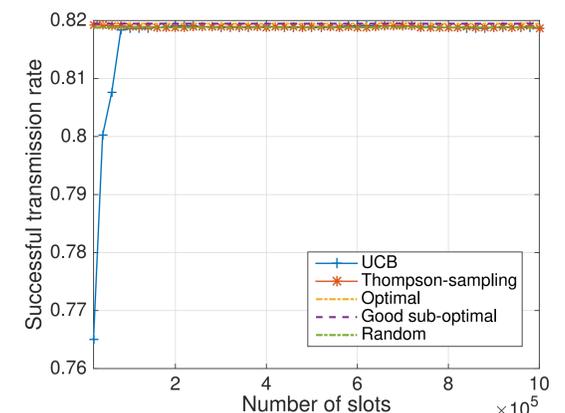
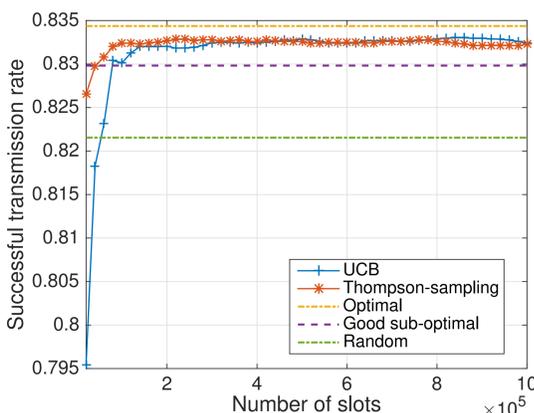
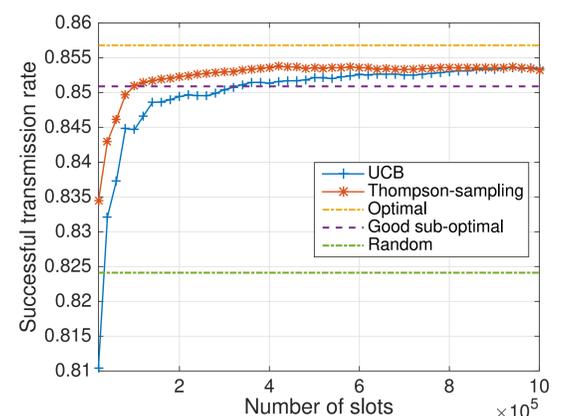
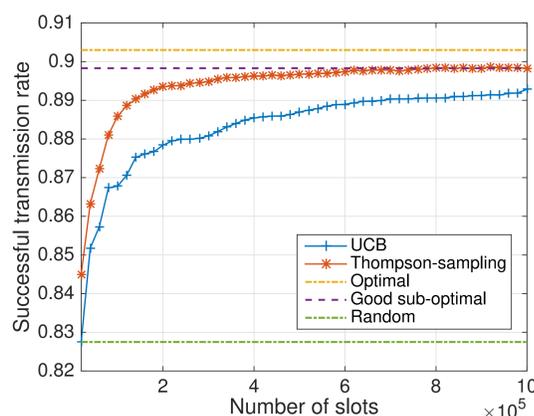


Figure 2: Performance of 2 MAB algorithms, compared to baseline algorithms (naive or optimal), when the proportion of dynamic end-devices in the network increases, for 10%, 30%, 50% and to 100% (limit scenario).

\Rightarrow Almost optimal performances!

\Rightarrow Very quick convergence!

6. NEAR OPTIMAL PERFORMANCES

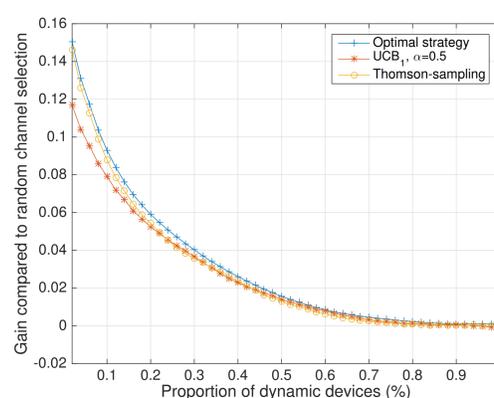


Figure 3: Learning with UCB₁ and TS, with more and more dynamic devices. \Rightarrow For any configuration, TS converges quickly to *near optimal* performances!

7. CONCLUSIONS

- Our approach is simple to set up: every dynamic object runs a simple on-line Multi-Armed Bandit *algorithm* to learn the *quality* of each channel, and aim at the most available channel
- *Economic*: low runtime complexity, low memory requirements
- In a *fully decentralized* manner, dynamic devices learn to fit in the channels almost optimally !
- *Convergence* is very *quick* to attain: about 50 communications for each device is enough !
- *Surprising result*: stochastic MAB algorithms also work very well in *non-stochastic environments* !

\Rightarrow With lots of dynamic objects in a IoT network, **using MAB learning helps to improve the successful transmission rate**, and increase *quality of service*.

8. MAIN REFERENCES

- [BBM⁺17] R. Bonnefoi, L. Besson, C. Moy, E. Kaufmann, and J. Palicot (2017). *Multi-Armed Bandit Learning in IoT Networks: Learning helps even in non-stationary settings*. Sent to the CrownCom 2017 conference in May 2017.
- [MPD16] C. Moy, J. Palicot, and S. J. Darak (2016). *Proof-of-Concept System for Opportunistic Spectrum Access in Multi-user Decentralized Networks*. *EAI Endorsed Transactions on Cognitive Communications*, 2.

MORE ON-LINE \rightarrow <http://lbo.k.vu/JdD2017>

9. THANKS TO ...

- ADDI association for the "PhD Students Day" 2017 !
- Magnet Inria team for Workshop on Decentralized ML 2017 !
- CentraleSupélec, IETR, Univ.Rennes 1, Inria (Lille) & CNRS
- Our advisors: Christophe Moy, Émilie Kaufmann & J. Palicot.