

Stratégies d'allocation des démodulateurs dans un réseau LoRaWAN

Alexandre Guitton, Megumi Kaneko, Nancy El Rachkidy

Inria

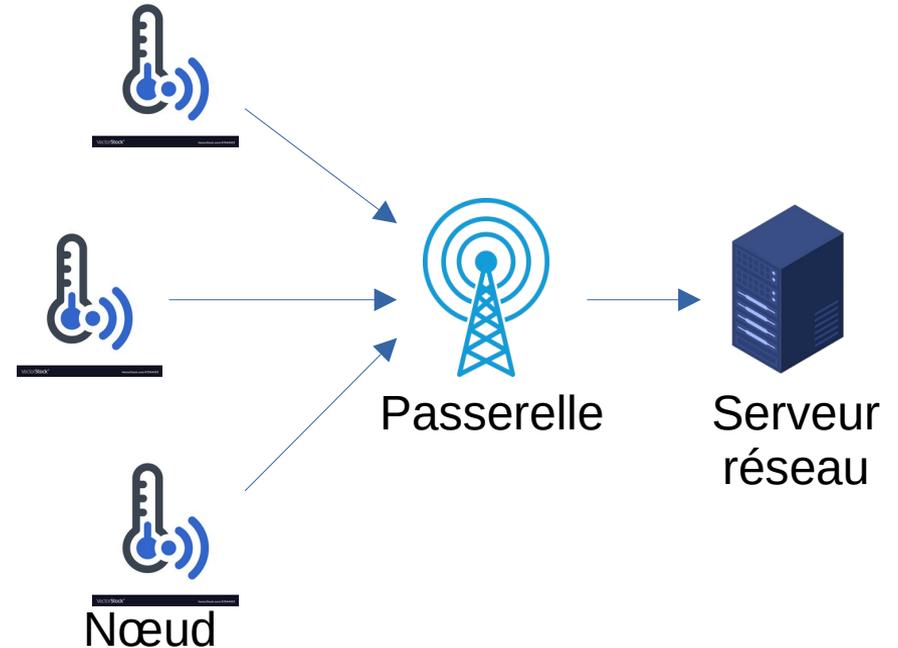
UCA
UNIVERSITÉ
Clermont
Auvergne

LIMOS

NII
大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

LoRaWAN et les démodulateurs

- Communications LoRaWAN :
nœuds → passerelles → serveur
réseau
- **Démodulateurs = circuits de la
passerelle pour démoduler des
trames**
 - Huit démodulateurs ⇒ huit trames
démodulables en parallèle (sur
des canaux différents et des SF
différents)



Problématique

- **Nombre limité de démodulateurs** (typiquement huit) ⇒ **performances LoRaWAN limitées**
- Quelques travaux proposent des **heuristiques** pour améliorer l'allocation des démodulateurs
 - A.Guitton and M.Kaneko. « Improving LoRa scalability by a recursive reuse of demodulators », Globecom 2020.
 - A.Guitton and M.Kaneko. « Multi-gateway demodulation in LoRa », Globecom 2022.
 - J.-T.Chen, M.Kaneko and A.Guitton. « An auction-based assignment method for LoRa multi-gateway networks », ICC 2023.
- **Quelles performances optimales peut-on atteindre ?**

Hypothèses

- Deux trames ne peuvent pas démarrer ou finir à exactement le même instant
- Le préambule des trames est détecté à l'instant de début du payload
- Pas de collision de trames

Notations

- F = ensemble de trames
 - Chaque trame $f \in F$ est transmise dans $[start(f);end(f)]$
- Pour un algorithme A :
 - $Gain(A(F))$ = nombre de trames démodulées (que l'on **souhaite maximiser**)
 - $Gain_{t1,t2}(A(F))$ = nombre de trames dont la démodulation finit dans $]t1;t2]$
 - A a un facteur d'approximation $r \geq 1$ si pour tout A^* et pour tout F ,
 $Gain(A^*(F))/Gain(A(F)) \leq r \Rightarrow A$ est au plus r fois pire que l'optimal

Algorithme Gloutonus

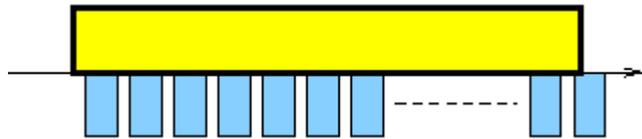
- Quand le préambule d'une nouvelle trame est détecté :
 - Si un démodulateur est libre \Rightarrow on démodule la trame
- Remarques :
 - Algorithme de type FIFO
 - C'est probablement l'algorithme implémenté dans les passerelles actuelles

Algorithme Prémptus

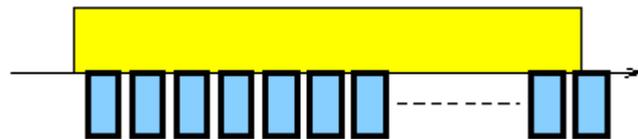
- Quand le préambule d'une nouvelle trame f est détecté :
 - Si un démodulateur est libre \Rightarrow on démodule f
 - Si un démodulateur est occupé avec une trame f_{old} telle que $end(f) < end(f_{old}) \Rightarrow$ on abandonne f_{old} et on démodule f (= préemption)
- Remarque : pour les résultats qui vont suivre, on doit considérer la trame f_{old} qui finit le plus tard, et non pas n'importe quelle trame qui finit après $end(f)$

Scénario 1 : 1 GW et 1 démod

- Théorème 1 : le facteur d'approximation de **Gloutonus** est 72...
 - Donc on pourrait démoduler jusqu'à 72 fois plus de trames que **Gloutonus** !
- Exemple :



Gain(Gloutonus)=1



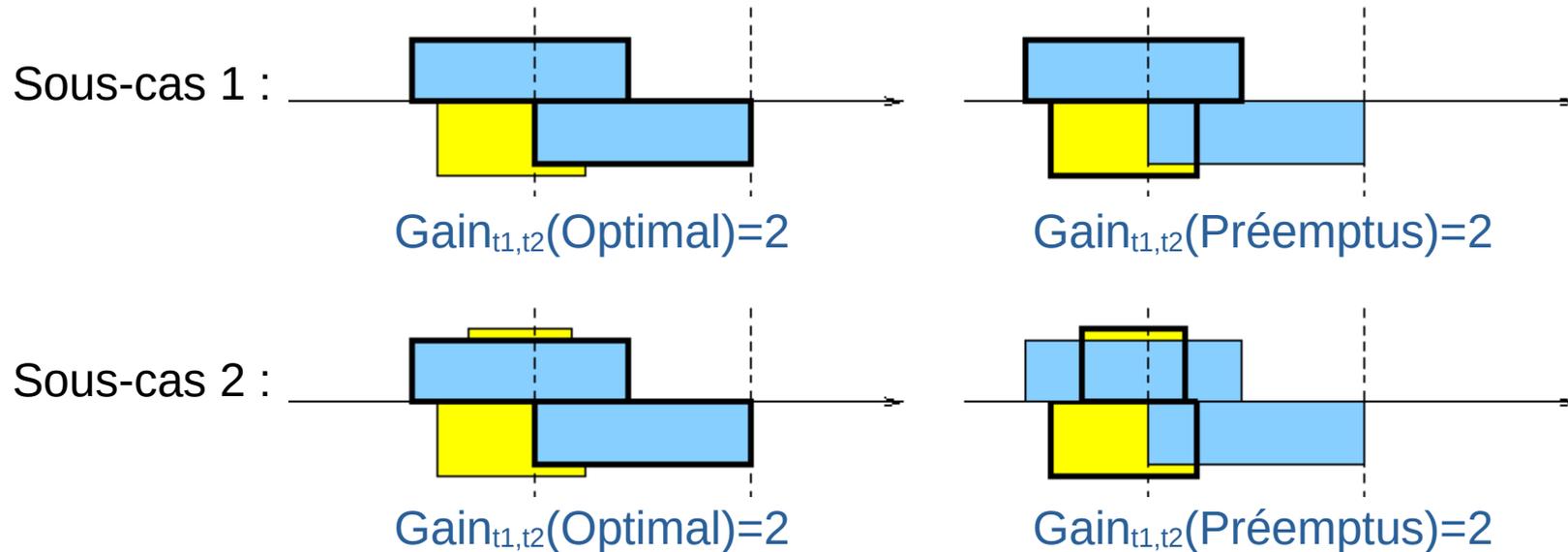
Gain(Optimal)=72

Scénario 1 : 1 GW et 1 démod

- Théorème 2 : *Préemptus* est optimal
- Sketch de la preuve :
 - Supposons qu'il existe un instant t tel que $Gain(OPT) > Gain(Préemptus)$
 - Donc à t , *OPT* vient de démoduler une trame f qui n'était pas choisie par *Préemptus*, avec $end(f) = t$
 - Le démodulateur de *Préemptus* était forcément occupé à $start(f)$ (sinon il aurait pris f), et il aurait dû abandonner sa trame pour prendre f qui termine plus tôt \Rightarrow contradiction

Scénario 2 : 1 GW et 2 démod

- Théorème 3 : **Préemptus** est optimal
- Exemples (deux sous-cas de la preuve)



Scénario 3 : 2 GW et 1 démod

- Chaque passerelle reçoit des trames qui peuvent être différentes (ou pas)
- Théorème 4 : **Préemptus** a un facteur d'approximation de 2
- **Peut-on imaginer des algorithmes de collaboration multi-passerelles pour améliorer ce facteur d'approximation ?**

Algorithme CollaboratusSimplus

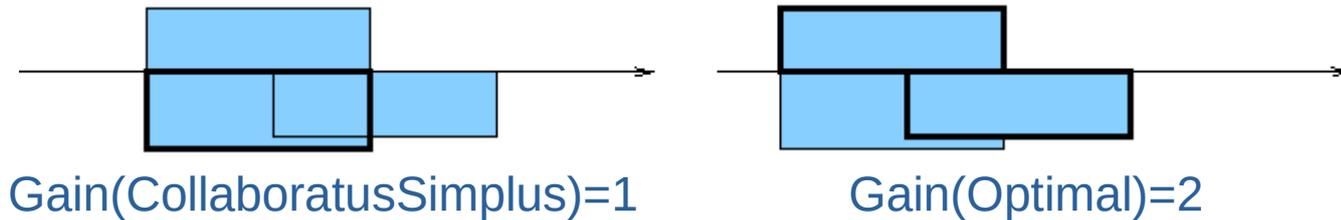
- Quand le préambule d'une nouvelle trame f est détecté :
 - On applique l'algorithme **Préemptus**
 - Si plusieurs passerelles envisagent de démoduler $f \Rightarrow$ seulement une passerelle (choisie aléatoirement) démodule f
- Remarque : on fait l'hypothèse que si GW1 détecte (le préambule de) f_1 et que GW2 détecte (le préambule de) f_2 , alors elles peuvent collaborer pour savoir si $f_1=f_2$, avant le début du payload

Alg. CollaboratusUnPeuComplicatus

- Similaire à **CollaboratusSimplus**, mais :
 - Quand une trame est reçue par plusieurs passerelles, on ne l'abandonne pas tout de suite \Rightarrow on l'abandonnera si une trame future est meilleure

Scénario 3 : 2 GW et 1 démod

- Théorème 5 : **CollaboratusSimplus** a un facteur d'approximation de 2



- Théorème 6 : **CollaboratusUnPeuComplicatus** a un facteur d'approximation de 1.5

Vérification par simulation

- Calcul de la solution optimale par ILP
- Ensembles F : trames aléatoires avec $SF \in [7;12]$ et taille variable pour le payload
- Résultat 1 :
 - Le gain de l'algorithme **Préemptus** est bien optimal pour 1 GW et 2 démodulateurs, sur 1000 instances testées
- Résultat 2 :
 - Les résultats obtenus par simulation respectent les bornes prouvées
 - En général, $\text{Gain}(\text{Gloutonus}) < \text{Gain}(\text{Préemptus}) < \text{Gain}(\text{CollaboratusSimplus}) < \text{Gain}(\text{CollaboratusUnPeuComplicatus})$
 - En général, tous les algorithmes (y compris **Gloutonus**) sont « bons »

Conclusion et perspectives

- L'allocation des démodulateurs dans LoRa est un problème intéressant
- Les passerelles rendent le problème compliqué
- Quelles performances peut-on espérer pour huit démodulateurs (ou plus) ?
- Quelles performances peut-on espérer pour plus de deux passerelles ?

Merci !

Questions ?